☆ THE ART OF COMPUTER PROGRAMMING ☆

☆☆☆ ERRATA TO VOLUME 2 (2nd edition) ☆☆☆

This document is a transcript of the notes I've made in my personal copy of *The Art of Computer Programming*, Volume 2, during the past 16 years. They apply to all copies of the second edition printed since January 1981.

Three levels of updates appear, "errors," "amendments," and "plans," indicated by three different typographic conventions:

▸**Page 666** line 1 —————————————— 04 Jul 1776

Technical or typographical errors (aka bugs) are the most critical items, so they are flagged with a '▸' preceding the page number. The date on which I first was told about the bug is shown; this is the effective date on which I paid the finder's fee. The necessary corrections are indicated in a straightforward way. If, for example, the book says 'n' where it should have said '$n+1$', the change is shown thus:

$n \rightsquigarrow n+1$

Page 666 line 2 —————————————— 14 Jul 1789

Amendments to the text appear in the same format as bugs, but without the '▸'. These are things I wish I had known about or thought of when I wrote the original text, so I added them later. The date is the date I drafted the new text.

Page 666 line 3 . . . . 20 Nov 1917

Plans for the future represent a third kind of item. In such notes I sketched my intentions about things that I wasn't ready to flesh out further when I wrote them down. You can identify these items because they're written in slanted type, and preceded by a bunch of dots '. . . . .' leading to the date on which I recorded the plan in my files.

A lot of material that I believe is interesting, important, and often beautiful, appears in this document. In other words, I think these pages contain a lot of good stuff. But I make no claim that the changes listed here will bring the books up to date. My shelves at home are bursting with preprints and reprints of significant research results that I want to digest and summarize, where appropriate, in the ultimate editions of Volumes 1–3. I'm not doing that now because I would surely have to do it over again later: New results continue to pour forth at a great rate, and I will have time to rewrite Volumes 1–3 only once. Volumes 4 and 5 need to be finished first. So I've put most of my effort so far into writing up those parts of the the total picture that seem to have converged to their near-final form. It follows, somewhat paradoxically, that the updates in this document are most current in the areas where there has been least activity.

On the other hand I do believe that the changes listed here bring Volume 2 completely up to date in two respects: (1) All of the research problems in the previous edition — i.e., all exercises that were rated 46 and above — have received new ratings of 45 or less whenever I learned of a solution; and in such cases, the answer now refers to that solution. Several of the research problems have also been rescinded or restated. (2) All of the historical information about pioneering developments has been amended whenever new details have come to my attention.

The ultimate, glorious, future editions of Volumes 1–3 are works in progress. Please let me know of any improvements that you think I ought to make. Send your comments either by snail mail to D. E. Knuth, Computer Science, Gates Building 4B, Stanford University, Stanford CA 94305-9045, or by email to `taocp@cs.stanford.edu`*. (Please, however, use that email address only to suggest improvements, not to ask technical questions.) Although I'm working full time on Volume 4 these days, I will try to reply to all such messages within a year of receipt.*

— Don Knuth, September 1997

Things have changed in the past two decades.

— BILL GATES (1995)

☆☆☆☆☆ SEMINUMERICAL ALGORITHMS ☆☆☆☆☆

Page v and throughout . . . . 20 Feb 1995

instead of using slanted type for emphasis, I will revert to the traditional style of italic type for emphasis, while retaining the slanted statements of theorems and names of publications

Page x line 15 —— 28 Feb 1997

book. $\leadsto$ book. If the rating number is divided by 5, the remainder indicates the amount of detailed work required. Thus, an exercise rated *24* may take longer to solve than an exercise that is rated *25*, but the latter will require more creativity.

Page xi replacement for exercise 3 —— 25 Mar 1995

3. [*34*] Leonhard Euler conjectured in 1772 that the equation $w^4+x^4+y^4=z^4$ has no solution in positive integers, but Noam Elkies proved in 1987 that infinitely many solutions exist [see *Math. Comp.* **51** (1988), 825–835]. Find all integer solutions such that $0 \le w \le x \le y < z < 10^6$.

4. [*M50*] Prove that when n is an integer, $n > 4$, the equation $w^n + x^n + y^n = z^n$ has no solution in positive integers w, x, y, z.

▸**Page xi** line −2 —— 31 Jul 1997

intrument $\leadsto$ *instrument*

▸**Page xii** line −6 —— 12 Apr 1981

300 $\leadsto$ 302

▸**Page 1** first line of first quote —— 16 Dec 1996

Anyone $\leadsto$ *Any one*

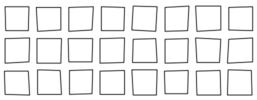
Page 1, line 4 from the bottom —— 10 Feb 1996

This is $\leadsto$ More importantly, they are crucial to the operation of *randomized algorithms*, which are often far superior to their deterministic counterparts. This is

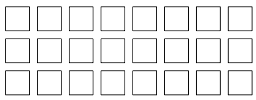
Page 2, line 8 —— 27 Apr 1996

f) *Recreation.* $\leadsto$

f) *Aesthetics.* A little bit of randomness makes computer-generated graphics and music seem more lively. For example, a pattern like



tends to look
more appealing than



in certain contexts. [See D. E. Knuth, *Bull. Amer. Math. Soc.* **1** (1979), 369.]

g) *Recreation.*

Page 2 line −2 — 13 Jul 1995

digits, and in 1955 $\leadsto$ digits. The Ferranti Mark I computer, first installed in 1951, had a built-in instruction that put 20 random bits into the accumulator using a resistance noise generator; this feature had been recommended by A. M. Turing. In 1955,

Page 3 top — 13 Jul 1995

line 4: 61; see also the review $\leadsto$ 61. See also S. H. Lavington's discussion of the Mark I in *CACM* **21** (1978), 4–12; the review
line 5: 43, $\leadsto$ 43;
lines 12–13: but this ... impractical $\leadsto$ as in the Ferranti Mark I, but this has proved to be unsatisfactory since it is impossible
line 14: and such $\leadsto$ moreover, such

Page 3 lines 15 and 16 — 22 Sep 1995

detect. ... methods led $\leadsto$ detect. Advances in technology made tables useful again during the 1990s, because a billion well-tested random bytes could be distributed on CDROM. George Marsaglia helped resuscitate random tables in 1995 by preparing a demonstration disk that contained 650 random megabytes, generated by combining the output of a noise-diode circuit with deterministically scrambled rap music. (He called it "white and black noise.")

The inadequacy of mechanical methods in the early days led

Page 3 lines −13 through −11 — 07 Aug 1995

usually called *pseudo-random* ... in this book $\leadsto$ often called *pseudorandom* or *quasirandom* sequences in the highbrow technical literature, but in most places of this book

Page 7 exercise 4 — 14 Apr 1995

line 1: [*10*] $\leadsto$ [*20*] (a)
line 2: zero? $\leadsto$ zero? (b) Use Table 1 to deduce what happens when Algorithm K is applied repeatedly with the starting value $X = 3830951656$.

▸**Page 7** first line of exercise 7 — 06 Mar 1981

least $\leadsto$ greatest

Page 8 new exercises — 14 Apr 1995

20. [*30*] Find all nonnegative $X < 10^{10}$ that lead ultimately via Algorithm K to the self-reproducing number in Table 1.

21. [*42*] Prove or disprove: The mapping $X \mapsto f(X)$ defined by Algorithm K has exactly five cycles, of respective lengths 3178, 1606, 1024, 943, and 1.

Page 8 new exercise — 13 Jul 1995

22. [*21*] (H. Rolletschek.) Would it be a good idea to generate random numbers by using the sequence $f(0)$, $f(1)$, $f(2)$, ..., where f is a random function, instead of using x_0, $f(x_0)$, $f(f(x_0))$, etc.?

Page 8 another new exercise ——— 26 Apr 1995

▸**23.** [*M26*] (D. Foata and A. Fuchs, 1970.) Show that each of the m^m functions $f(x)$ considered in exercise 6 can be represented as a sequence $(x_0, x_1, \ldots, x_{m-1})$ having the following properties:

(1) $(x_0, x_1, \ldots, x_{m-1})$ is a permutation of $\bigl(f(0), f(1), \ldots, f(m-1)\bigr)$.

(2) $\bigl(f(0), \ldots, f(m-1)\bigr)$ can be uniquely reconstructed from $(x_0, x_1, \ldots, x_{m-1})$.

(3) The elements that appear in cycles of f are $\{x_0, x_1, \ldots, x_{k-1}\}$, where k is the largest subscript such that these k elements are distinct.

(4) $x_j \notin \{x_0, x_1, \ldots, x_{j-1}\}$ implies $x_{j-1} = f(x_j)$, unless x_j is the smallest element in a cycle of f.

(5) $\bigl(f(0), f(1), \ldots, f(m-1)\bigr)$ is a permutation of $(0, 1, \ldots, m-1)$ if and only if $(x_0, x_1, \ldots, x_{m-1})$ represents the *inverse* of that permutation by the "unusual correspondence" of Section 1.3.3.

(6) $x_0 = x_1$ if and only if $(x_1, \ldots, x_{m-1})$ represents an oriented tree by the construction of exercise 2.3.4.4–18, with $f(x)$ the parent of x.

Page 9 line −3 ——— 16 Mar 1982

$f(X_n)$; $\rightsquigarrow$ $f(X_n)$, when f transforms a finite set into itself;

Page 12 lines 6 and 7 after (2) ——— 06 Oct 1996

It is convenient … $w+1$; this will happen if $\rightsquigarrow$ We could represent w by 0, since (2) will not normally be used when $X = 0$; but it is most convenient … $w + 1$. This will happen, and overflow will be avoided, if

Page 15 replacement for exercise 3 ——— 27 Aug 1995

▸**3.** [*M25*] Many computers do not provide the ability to divide a two-word number by a one-word number; they provide only operations on single-word numbers, such as $\operatorname{himult}(x,y) = \lfloor xy/w \rfloor$ and $\operatorname{lomult}(x,y) = xy \bmod w$, when x and y are nonnegative integers less than the word size w. Explain how to evaluate $ax \bmod m$ in terms of himult and lomult, assuming that $0 \le a, x < m < w$ and that $m \perp w$. You may use precomputed constants that depend on a, m, and w.

Page 15 replacement for exercise 9 ——— 30 Aug 1995

▸**9.** [*M25*] Most high-level programming languages fail to provide a good way to divide a two-word integer by a one-word integer, nor do they provide the himult operation of exercise 3. The purpose of this exercise is to find a reasonable way to cope with such limitations when we wish to evaluate $ax \bmod m$ for variable x and for constants $0 < a < m$.

a) Prove that if $q = \lfloor m/a \rfloor$, we have $a\bigl(x - (x \bmod q)\bigr) = \lfloor x/q \rfloor \bigl(m - (m \bmod a)\bigr)$.

b) Use the identity of (a) to evaluate $ax \bmod m$ without computing any numbers that exceed m in absolute value, assuming that $a^2 \le m$.

c) The solution to (b) sometimes works also when $a^2 > m$. Exactly how many multipliers a are there for which the intermediate results in that method never exceed m, for all x between 0 and m?

10. [*M30*] Continuing exercise 9, show that it is possible to evaluate $ax \bmod m$ using only the following basic operations:

(i) $u \times v$, where $u \ge 0$, $v \ge 0$, and $uv \le m$;

(ii) $\lfloor u/v \rfloor$, where $0 < v \le u \le m$;

(iii) $(u - v) \bmod m$, where $0 \le u, v < m$.

In fact, it is always possible to do this with at most 12 operations of types (i) and (ii), and with a bounded number of operations of type (iii), not counting the precomputation of constants that depend on a and m. For example, explain how to proceed when a is 62089911 and m is $2^{31} - 1$. (These constants appear in Table 3.3.4–1.)

Page 15 new exercises — 16 Sep 1995

▸**11.** [*M23*] a) What's a good way to multiply a given 10-digit number by 10, modulo 9999998999, using pencil and paper or an abacus?

b) Same question, but multiply instead by 999999900 (modulo 9999998999).

c) Explain how to compute the powers $999999900^n \bmod 9999998999$, for $n = 1, 2, 3, \ldots$.

d) Relate such computations to the decimal expansion of 1/9999998999.

e) Show that these ideas make it possible to implement certain kinds of linear congruential generators that have extremely large moduli, using only a few operations per generated number.

12. [*M25*] Repeat the previous exercise, but with modulus 9999999001 and with multipliers 10 and 8999999101.

13. [*M25*] Generalize the ideas of the previous two exercises, obtaining a large family of linear congruential generators with extremely large moduli.

Page 19 line −12 — 20 Oct 1995

[R. D. Carmichael, ... 238.] $\leadsto$ [C. F. Gauss, *Disquisitiones Arithmeticæ* (1801), §90–92.]

Page 22 new exercise — 19 Sep 1995

▸**22.** [*M25*] Discuss the problem of finding moduli $m = b^k \pm b^l \pm 1$ so that the subtract-with-borrow and add-with-carry generators of exercise 3.2.1.1–13 will have very long periods.

Page 24 line 2 of exercise 3 — 19 Sep 1995

$2^{23} + 2^{14} + 2^2 + 1 \leadsto 2^{23} + 2^{13} + 2^2 + 1$

Page 25 line 4 of exercise 8 — 25 Jun 1982

equal probability $\leadsto$ nearly equal frequency

▸**Page 26** line 11 — 30 Sep 1982

in exercise 3.3.4–25. $\leadsto$ in Coveyou's paper cited in the answer to exercise 8.

Page 26 line 3 before (5) — 15 Jul 1995

(X_n, X_{n+1}). $\leadsto$ (X_n, X_{n+1}). John Mauchly, in an unpublished paper presented to a statistics conference in 1949, extended the middle square method by using the recurrence $X_n = \text{middle}\,(X_{n-1} \cdot X_{n-6})$.

Page 27 lines 5–7 — 15 Jul 1995

Exercise 11 ... $m = 2^e$. $\leadsto$ Exercise 30 proves that (7) has a period of length $2^{e-1}(2^{55} - 1)$ when $m = 2^e$.

Page 27 bottom ________ 12 Aug 1995

lines −8 and −7: yet; and ... 1958. $\rightsquigarrow$ yet, except in exercise 3.2.1.2–22.
line −3: The only ... sequence (7) $\rightsquigarrow$ The main reason it is difficult to recommend sequences like (7)
lines −2 and −1: it does or does not $\rightsquigarrow$ they do or do not

Page 28 replacement for Table 1 ________ 18 Aug 1995

Table 1

LAGS THAT YIELD LONG PERIODS MOD 2

$(24, 55)$	$(37, 100)$	$(83, 258)$	$(273, 607)$	$(576, 3217)$	$(7083, 19937)$
$(38, 89)$	$(30, 127)$	$(107, 378)$	$(1029, 2281)$	$(4187, 9689)$	$(9739, 23209)$

For extensions of this table, see N. Zierler and J. Brillhart, *Information and Control* **13** (1968), 541–554, **14** (1969), 566–569, **15** (1969), 67–69; Y. Kurita and M. Matsumoto, *Math. Comp.* **56** (1991), 817–821; Heringa, Blöte, and Compagner, *Int. J. Mod. Phys.* **C3** (1992), 561–564.

Page 28 lines 5–12 after Table 1 ________ 18 Aug 1995

The fact that ... additive sequences, $\rightsquigarrow$ The numbers 24 and 55 in (7) are commonly called *lags*, and the numbers X_n defined by (7) are said to form a *lagged Fibonacci sequence*. Lags like $(24, 55)$ work well because of theoretical results developed in some of the exercises below. It is of course better to use somewhat larger lags if your application happens to use, say, groups of 55 values at a time; the numbers generated by (7) will never have X_n lying strictly between X_{n-24} and X_{n-55} (see exercise 2). J.-M. Normand, H. J. Herrmann, and M. Hajjar detected slight biases in the numbers generated by (7) when they did extensive high-precision Monte Carlo studies requiring 10^{11} random numbers [*J. Statistical Physics* **52** (1988), 441–446]; but larger values of k decreased the bad effects. Table 1 lists several useful pairs (l, k) for which the sequence $X_n = (X_{n-l} + X_{n-k}) \bmod 2^e$ has period length $2^{e-1}(2^k - 1)$. The case $(l, k) = (30, 127)$ should be large enough for most applications, especially in combination with other randomness-enhancing techniques that we will discuss later.

George Marsaglia [*Comp. Sci. and Statistics: Symposium on the Interface* **16** (1984), 3–10] has suggested replacing (7) by

$$X_n = (X_{n-24} \cdot X_{n-55}) \bmod m, \qquad n \ge 55, \tag{7$'$}$$

where m is a multiple of 4 and where X_0 through X_{54} are odd, not all congruent to 1 (modulo 4). Then the second-least significant bits have a period of $2^{55} - 1$, while the most significant bits are more thoroughly mixed than before since they depend on all bits of X_{n-24} and X_{n-55} in an essential way. Exercise 31 shows that the period length of sequence (7′) is only slightly less than that of (7).

Lagged Fibonacci generators have been used successfully in many situations since 1958, so it came as shock to discover in the 1990s that they actually fail an extremely simple, non-contrived test for randomness (see exercise 3.3.2–31). A "workaround" that avoids such problems by discarding appropriate elements of the sequence is described near the end of this section.

Instead of considering purely additive or purely multiplicative sequences,

▸**Page 29** line 24 — 03 Apr 1985

3.3.4–26 ⇝ 3.3.4–24

Page 30 middle — 26 Feb 1996

line 22: ≤ 100 ⇝ ≤ 168
line 23: E. J. Watson ... 369. ⇝ W. Stahnke, *Math. Comp.* **27** (1973), 977–980.

▸**Page 30** line 25 — 15 Jul 1995

3.3.4–26 ⇝ 3.3.4–24

Page 30 line −10 — 12 Aug 1995

listing of all the ⇝ listing of certain

Page 30 lines −5 and −4 — 12 Aug 1995

(See ... method.) ⇝ Subsequent experiments have shown that (7) gives better results.

Page 31 lines 3–5 after (11) — 16 Apr 1995

k; his method ... appear in ⇝ k. His method is easy to state (exercise 17) and reasonably efficient to program (exercise 29), but it is unsuitable for random number generation because it changes the value of $X_{n-1}+\cdots+X_{n-k}$ very slowly: All k-tuples occur, but not in a very random order. A better class of functions f that yield the maximum period m^k is considered in

Page 31 new material preceding line −18 — 18 Sep 1995

Many other schemes have been proposed for random number generation, and the most interesting of these alternative methods may well be the *inversive congruential sequences* suggested by Eichenauer and Lehn [*Statistische Hefte* **27** (1986), 315–326]:

$$X_{n+1} = (aX_n^{-1} + c) \bmod p\,. \tag{12}$$

Here p is prime, X_n ranges over the set $\{0, 1, \ldots, p-1, \infty\}$, and inverses are defined by $0^{-1} = \infty$, $\infty^{-1} = 0$, otherwise $X^{-1}X \equiv 1$ (modulo p). Since 0 is always followed by ∞ and then by c in this sequence, we could simply define $0^{-1} = 0$ for purposes of implementation; but the theory is cleaner and easier to develop when $0^{-1} = \infty$. Efficient algorithms suitable for hardware implementation are available for computing X^{-1} modulo p; see, for example, exercise 4.5.2–35. Unfortunately, however, this operation is not in the repertoire of most computers. Exercise 35 shows that many choices of a and c yield the maximum period length $p+1$. Exercise 37 demonstrates the most important property: Inversive congruential sequences are completely free of the lattice structure that is characteristic of linear congruential sequences.

Page 31 line −14 — 16 Apr 1995

unjustified (although we believe it is), so ⇝ unjustified — indeed, they may be right — so

Page 31 replacement for the last eight lines ______ 15 Jun 1983

random sequence can, for example, be used to permute the elements of another, as suggested by M. D. MacLaren and G. Marsaglia [*JACM* **12** (1965), 83–89; cf. Marsaglia and Bray, *CACM* **11** (1968), 757–759]:

Page 32 and page 33, displayed equations ______ 18 Sep 1995

(12) $\rightsquigarrow$ (13) (13) $\rightsquigarrow$ (14)

Page 32 lines 9 and 10 after (12) ______ 18 Aug 1995

In particular, ... 2^{35}. $\rightsquigarrow$ In particular, if we reject the value 0 when it occurs in the Y sequence, so that $\langle Y_n \rangle$ has period length $2^{35} - 1$, the numbers generated by Algorithm M from (12) will have a period of length $2^{70} - 2^{35}$. [See J. Arthur Greenwood, *Comp. Sci. and Statistics: Symp. on the Interface* **9** (1976), 222.]

Page 33 new paragraphs before the exercises ______ 26 Aug 1995

Shuffling methods have an inherent defect, however: They change only the order of the generated numbers, not the numbers themselves. For most purposes the order is the critical thing, but if a random number generator fails the “birthday spacings” test discussed in Section 3.3.2 or the random-walk test of exercise 3.3.2–31 it will not fare much better after it has been shuffled. Shuffling also has the comparative disadvantage that it does not allow us to start at a given place in the period, or to skip quickly from X_n to X_{n+k} for large k.

Many people have therefore suggested combining two sequences $\langle X_n \rangle$ and $\langle Y_n \rangle$ in a much simpler way, which avoids both of the defects of shuffling: We can use a combination like

$$Z_n = (X_n - Y_n) \bmod m \tag{15}$$

when $0 \le X_n < m$ and $0 \le Y_n < m' \le m$. Exercises 13 and 14 discuss the period length of such sequences; exercise 3.3.2–23 shows that (15) tends to enhance the randomness when the seeds X_0 and Y_0 are chosen independently.

An even simpler way to remove the structural biases of arithmetically generated numbers was proposed already in the early days of computing by J. Todd and O. Taussky Todd [*Symp. on Monte Carlo Methods* (Wiley, 1956), 15–28]: We can just throw away some numbers of the sequence. Their suggestion was of little use with linear congruential generators, but it has become quite appropriate nowadays in connection with generators like (7) that have extremely long periods because we have plenty of numbers to discard.

The simplest way to improve the randomness of (7) is to use only every jth term, for some small j. But a better scheme, which may be even simpler, is to use (7) to produce say 500 random numbers in an array and to use only the first 55 of them. After those 55 have been consumed, we generate 500 more in the same way. This idea was proposed by Martin Lüscher [*Computer Physics Communications* **79** (1994), 100–110], based on the theory of chaos in dynamical systems: We can regard (7) as a process that maps 55 values $(X_{n-55}, \ldots, X_{n-1})$ into another vector of 55 values $(X_{n+t-55}, \ldots, X_{n+t-1})$. Suppose we generate $t \ge 55$ values and use the first 55 of these. Then if $t = 55$ the new vector of values is rather close to the old; but if $t \approx 500$ there is almost no correlation between old and new (see exercise 33). For the analogous case of add-with-carry or subtract-with-borrow generators (exercise 3.2.1.1–13), the vectors are in fact known to be the radix-b representation of numbers in a linear congruential generator, and the relevant multiplier when we generate t numbers at a time is b^{-t}. Lüscher's theory for this case can therefore be confirmed with the spectral test of Section 3.3.4.

Page 33 additional new material before the exercises ———— 08 Aug 1995

(a small vertical space will separate this from the previous paragraph)

Random number generators typically do only a few multiplications and/or additions to get from one element of the sequence to the next. When such generators are combined in various ways, common sense tells us that the resulting sequences ought to be indistinguishable from truly random numbers. But intuitive hunches are no substitute for rigorous mathematical proof. If we are willing to do more work — say 1000 or 1000000 times as much — we can obtain sequences for which substantially better theoretical guarantees of randomness are available.

For example, consider the sequence of bits B_1, B_2, ... generated by

$$X_{n+1} = X_n^2 \bmod M, \qquad B_n = X_n \bmod 2, \tag{16}$$

[Blum, Blum, and Shub, *SICOMP* **15** (1986), 364–383], or the more elaborate sequence generated by

$$X_{n+1} = X_n^2 \bmod M, \qquad B_n = X_n \cdot Z \bmod 2, \tag{17}$$

where the dot product of r-bit binary numbers $(x_{r-1} \ldots x_0)_2$ and $(z_{r-1} \ldots z_0)_2$ is $x_{r-1}z_{r-1} + \cdots + x_0 z_0$; here Z is an r-bit "mask," and r is the number of bits in M. The modulus M should be the product of two large primes of the form $4k+3$, and the starting value X_0 should be relatively prime to M. Rule (17), suggested by Leonid Levin, is a take-off on von Neumann's original middle-square method; we will call it the *muddle-square method*, because it jumbles the bits of the squares. Rule (16) is, of course, the special case $Z = 1$.

Section 3.5F contains a proof that, when X_0, Z, and M are chosen at random, the sequences generated by (16) and (17) pass all statistical tests for randomness that require no more work than factoring large numbers. In other words, the bits cannot be distinguished from truly random numbers by any computation lasting less than 100 years on today's fastest computers, when M is suitably large, unless it is possible to find the factors of a nontrivial fraction of such numbers much more rapidly than is presently known. Formula (16) is simpler than (17), but the modulus M in (16) has to be considerably larger than it does in (17) if we want to achieve the same statistical guarantees.

Page 34 line 7 of exercise 8 ———— 15 Jul 1995

either $d \equiv 0$ *or both* $a \equiv 1$ *and* $cd \equiv 6 \rightsquigarrow d \not\equiv 3c$

Page 37 new exercise ——— 16 Apr 1995

▸**29.** [*40*] Find a good way to compute the function $f(x_1,\ldots,x_k)$ defined by Martin's sequence in exercise 17, given only the k-tuple $(x_1,\ldots,x_k)$.

Page 37 new exercise ——— 15 Jul 1995

30. [*M37*] (R. P. Brent.) Let $f(x)=x^k-a_1x^{k-1}-\cdots-a_k$ be a primitive polynomial modulo 2, and suppose that $X_0,\ \ldots,\ X_{k-1}$ are integers not all even. (a) Prove that the period of the recurrence $X_n=(a_1X_{n-1}+\cdots+a_kX_{n-k}) \bmod 2^e$ is $2^{e-1}(2^k-1)$ for all $e\geq 1$ if and only if $f(x)^2+f(-x)^2\not\equiv 2f(x^2)$ and $f(x)^2+f(-x)^2\not\equiv 2(-1)^kf(-x^2)$ (modulo 8). [*Hint:* We have $x^{2^k}\equiv -x$ (modulo 4 and $f(x)$) if and only if $f(x)^2+f(-x)^2\equiv 2f(x^2)$ (modulo 8).] (b) Prove that this condition always holds when $f(x)=x^k\pm x^l\pm 1$ is primitive modulo 2 and $k>2$.

Page 37 new exercise ——— 18 Aug 1995

31. [*M30*] (G. Marsaglia.) What is the period length of the sequence (7′) when $m=2^e\geq 8$? Assume that $X_0,\ \ldots,\ X_{54}$ are not all $\equiv\pm 1$ (modulo 8).

Page 37 new exercises ——— 26 Sep 1995

32. [*M21*] What recurrences are satisfied by the elements of the subsequences $\langle X_{2n}\rangle$ and $\langle X_{3n}\rangle$, when $X_n=(X_{n-24}+X_{n-55}) \bmod m$?

▸**33.** [*M23*] (a) Let $g_n(z)=X_{n+30}+X_{n+29}z+\cdots+X_nz^{30}+X_{n+54}z^{31}+\cdots+X_{n+31}z^{54}$, where the X's satisfy the lagged Fibonacci recurrence (7). Find a simple relation between $g_n(z)$ and $g_{n+t}(z)$. (b) Express X_{500} in terms of $X_0,\ \ldots,\ X_{54}$.

Page 37 new exercises ——— 18 Sep 1995

34. [*M25*] Prove that the inversive congruential sequence (12) has period $p+1$ if and only if the polynomial $f(x)=x^2-cx-a$ has the following two properties: (i) $x^{p+1} \bmod f(x)$ is a nonzero constant, when computed with polynomial arithmetic modulo p; (ii) $x^{(p+1)/q} \bmod f(x)$ has degree 1 for every prime q that divides $p+1$. [*Hint:* Consider powers of the matrix $\binom{0\ 1}{a\ c}$.]

35. [*HM30*] How many pairs (a,c) satisfy the conditions of exercise 34?

36. [*M25*] Prove that the inversive congruential sequence $X_{n+1}=(aX_n^{-1}+c) \bmod 2^e$, $X_0=1$, $e\geq 3$, has period length 2^{e-1} whenever $a \bmod 4=1$ and $c \bmod 4=2$.

▸**37.** [*HM32*] Let p be prime and assume that $X_{n+1}=(aX_n^{-1}+c) \bmod p$ defines an inversive congruential sequence of period $p+1$. Also let $0\leq b_1<\cdots<b_d\leq p$, and consider the set

$$V=\{(X_{n+b_1},X_{n+b_2},\ldots,X_{n+b_d}) \mid 0\leq n\leq p \text{ and } X_{n+b_j}\neq\infty \text{ for } 1\leq j\leq d\}.$$

This set contains $p+1-d$ vectors, any d of which lie in some $(d-1)$-dimensional hyperplane $H=\{(v_1,\ldots,v_d) \mid r_1v_1+\cdots+r_dv_d\equiv r_0 \text{ (modulo } p)\}$, where $(r_1,\ldots,r_d)\not\equiv(0,\ldots,0)$. Prove that no $d+1$ vectors of V lie in the same hyperplane.

▸**Page 41** rounding errors in Table 1 ——— 15 Jan 1983

$\nu=2$, $p=25\%$: 0.5753 ⇝ 0.5754
$\nu=6$, $p=1\%$: 0.8720 ⇝ 0.8721
$\nu=11$, $p=99\%$: 24.73 ⇝ 24.72
$\nu=12$, $p=75\%$: 14.84 ⇝ 14.85
x_p, $p=25\%$: −.675 ⇝ −.674
x_p, $p=75\%$: .675 ⇝ .674

▸**Page 45** line −9 ——— 27 Jan 1981

though though $\leadsto$ though

▸**Page 54** numerator of Eq. (20) ——— 28 Jul 1981

$\int_{(z_1,\ldots,z_k)\text{ in }S\text{ and }z_1+\cdots+z_k\le v} \leadsto \int_{(z_1,\ldots,z_k)\text{ in }S\text{ and }z_1^2+\cdots+z_k^2\le v}$

▸**Page 55** line 10 ——— 02 Feb 1981

0 and 1 $\leadsto$ 0 and n

Page 55 replacement for the last 9 lines, through page 56 line 5 ——— 13 Aug 1996

general formulas

$$\Pr\left(K_n^+ \le \frac{t}{\sqrt{n}}\right) = \frac{t}{n^n}\sum_{0\le k\le t}\binom{n}{k}(k-t)^k(t+n-k)^{n-k-1} \tag{25}$$

$$= 1-\frac{t}{n^n}\sum_{t<k\le n}\binom{n}{k}(k-t)^k(t+n-k)^{n-k-1}. \tag{26}$$

The distribution of K_n^- is exactly the same. Equation (26) was first obtained by N. V. Smirnov [*Uspekhi Mat. Nauk* **10** (1944), 176–206]; see also Z. W. Birnbaum and Fred H. Tingey, *Annals Math. Stat.* **22** (1951), 592–596. Smirnov derived the asymptotic formula

$$\Pr(K_n^+\le s) = 1-e^{-2s^2}\bigl(1-\tfrac{2}{3}s/\sqrt{n}+O(1/n)\bigr) \tag{27}$$

for all fixed $s\ge 0$; this yields the approximations for large n that appear in Table 2.

Abel's binomial theorem, Eq. 1.2.6–(16), shows the equivalence of (25) and (26). We can extend Table 2 using either formula, but there is an interesting tradeoff: Although the sum in (25) has only about $s\sqrt{n}$ terms, when $s=t/\sqrt{n}$ is given, it must be evaluated with multiple-precision arithmetic, because the terms are large and their leading digits cancel out. No such problem arises in (26), since its terms are all positive; but (26) has $n-s\sqrt{n}$ terms.

▸**Page 58** exercise 19 ——— 27 Jan 1981

Kolomogrov $\leadsto$ Kolmogorov

Page 58 exercise 20 ——— 13 Aug 1996

(28) $\leadsto$ (27)

Page 59 new exercise ——— 20 Aug 1995

25. [*HM26*] Suppose $Y_i=\sum_{j=1}^n a_{ij}X_j+\mu_i$ for $1\le i\le m$, where $X_1,\ldots,X_n$ are independent random variables with mean zero and unit variance, and the matrix $A=(a_{ij})$ has rank n. (a) Express the covariance matrix $C=(c_{ij})$, where $c_{ij}=\mathrm{E}(Y_i-\mu_i)(Y_j-\mu_j)$, in terms of the matrix A. (b) Prove that if $\overline{C}=(\bar c_{ij})$ is any matrix such that $C\overline{C}C=C$, the statistic $V=\sum_{i=1}^m\sum_{j=1}^m(Y_i-\mu_i)(Y_j-\mu_j)\bar c_{ij}$ is equal to $X_1^2+\cdots+X_n^2$. [Consequently, if the X_j have the normal distribution, V has the chi-square distribution with n degrees of freedom.]

Page 59 near the top — 18 Aug 1995

line 6: tests that have been $\leadsto$ tests that have traditionally been
line 13: here.) $\leadsto$ here. Indeed, this section introduces several things that will be important to us later in quite different contexts.)

▸**Page 60** lower middle — 20 Aug 1995

line −15: tests. Alternatively, ... and $\leadsto$ tests, remembering that the tests aren't independent of each other. Alternatively, George Marsaglia has proved that
lines −12 to −9: $V_2 - 2V_1$ should ... 264.) $\leadsto$ $V_2 - V_1$ should have the chi-square distribution with $d(d-1)$ degrees of freedom when n is large. (See exercise 24.)

Page 62 lines 12–16 — 21 Dec 1989

5 different $\leadsto$ 5 values
⋮
2 different $\leadsto$ 2 values
1 different $\leadsto$ 1 value

Page 63 line −2 — 05 Nov 1996

the paper by Hermann $\leadsto$ the papers by George Pólya, *Zeitschrift für angewandte Math. und Mech.* **10** (1930), 96–97; Hermann

Page 64 line 4 after Algorithm P — 09 Sep 1980

exchange $U_r U_s$ $\leadsto$ exchange $U_r \leftrightarrow U_s$

▸**Page 64** line −11 — 02 Jul 1997

$U_r\ U_s \leadsto U_r \leftrightarrow U_{s+1}$

Page 65 in (10) — 17 Jul 1995

$$\frac{1}{n} \leadsto \frac{1}{n-6}$$

▸**Page 66** left side of second equation in (14) — 02 Feb 1981

$Z_{pj} \leadsto Z_{qj}$

▸**Page 67** right side of second equation in (18) — 26 Mar 1981

$Z_{pj} \leadsto Z_{qj}$

Page 68 replacement for lines 5–7 following (22) — 17 Jul 1995

The true inverse, A, is $n^{-1}C_1^{-1} - n^{-2}C_1^{-1}C_2C_1^{-1} + n^{-3}C_1^{-1}C_2C_1^{-1}C_2C_1^{-1} - \cdots$, and it turns out that $C_1^{-1}C_2C_1^{-1}$ is very nearly equal to $-6C^{-1}$. Therefore $A \approx C_1^{-1}/(n-6)$.

Page 70 line 17 — 07 Sep 1995

J. Serial correlation test. $\leadsto$ **J. Birthday spacings test.** George Marsaglia introduced a new kind of test in 1984: We throw n balls into m urns as in the collision test, but now we think of the urns as "days of a year" and the balls as "birthdays." Suppose the birthdays are $(Y_1, \ldots, Y_n)$, where $0 \le Y_k < m$. Sort them into nondecreasing order $Y_{(1)} \le \cdots \le Y_{(n)}$; then define n "spacings" $S_1 = Y_{(2)} - Y_{(1)}, \ldots, S_{n-1} = Y_{(n)} - Y_{(n-1)}$, $S_n = Y_{(1)} + m - Y_{(n)}$; finally sort the spacings into order, $S_{(1)} \le \cdots \le S_{(n)}$. Let R be the number of equal spacings, i.e., the number of indices j such that $1 < j \le n$ and $S_{(j)} = S_{(j-1)}$. When $m = 2^{25}$ and $n = 512$, we should have

$R =$	0	1	2	≥ 3
with probability	.368801577	.369035243	.183471182	.078691997

(The average number of equal spacings for this choice of m and n should be approximately 1.) Repeat the test 1000 times, say, and do a chi-square test with 3 degrees of freedom to compare the empirical R's with the correct distribution; this will tell whether or not the generator produces reasonably random birthday spacings. Exercises 28–30 develop the theory behind this test and formulas for other values of m and n.

Such a test of birthday spacings is important primarily because of the remarkable fact that lagged Fibonacci generators consistently *fail* it, although they pass the other traditional tests quite nicely. [Dramatic examples of such failures were reported by Marsaglia, Zaman, and Tsang, *Stat. and Prob. Letters* **8** (1990), 35–39.] Consider, for example, the sequence

$$X_n = (X_{n-24} + X_{n-55}) \bmod m$$

of Eq. 3.2.2–(7). The numbers of this sequence satisfy

$$X_n + X_{n-86} \equiv X_{n-24} + X_{n-31} \pmod{m}$$

because both sides are congruent to $X_{n-24} + X_{n-55} + X_{n-86}$. Therefore two pairs of differences are equal: $X_n - X_{n-24} \equiv X_{n-31} - X_{n-86}$, and $X_n - X_{n-31} \equiv X_{n-24} - X_{n-86}$. Whenever X_n is reasonably close to X_{n-24} or X_{n-31} (as it should be in a truly random sequence), the difference has a good chance of showing up in two of the spacings. So we get significantly more cases of equality—typically $R \approx 2$ on the average, not 1. But if we discount from R any equal spacings that arise from the stated congruence, the resulting statistic R' usually does pass the birthday test. (One way to avoid failure is to discard certain elements of the sequence, using for example only X_0, X_2, X_4, … as random numbers; then we never get all four elements of the set $\{X_n, X_{n-24}, X_{n-31}, X_{n-86}\}$, and the birthday spacings are no problem. An even better way to avoid the problem is to discard consecutive batches of numbers as suggested by Lüscher; see Section 3.2.2.) Similar remarks apply to the subtract-with-borrow and add-with-carry generators of exercise 3.2.1.1–13.

K. Serial correlation test.

Page 71 replacement for lines 3–14 ________ 18 Jul 1995

$$\mu_n = \frac{-1}{n-1}, \qquad \sigma_n^2 = \frac{n^2}{(n-1)^2(n-2)}, \qquad n > 2. \tag{25}$$

We expect C to be between these limits about 95 percent of the time.

The formula for σ_n^2 in (25) is an upper bound valid for serial correlations between independent random variables from an arbitrary distribution. When the U's are uniformly distributed, the true variance is obtained by subtracting $\frac{24}{5}n^{-2} + O(n^{-7/3}\log n)$. (See exercise 20.)

Page 71 line −17 ________ 07 Sep 1995

K. Tests on subsequences. $\rightsquigarrow$ **L. Tests on subsequences.**

Page 72 line 1 ________ 07 Sep 1995

L. Historical remarks $\rightsquigarrow$ **M. Historical remarks**

▸**Page 72** line 12 ________ 08 Jun 1996

1–4 $\rightsquigarrow$ 1–5

Page 73 line 12 ________ 23 Sep 1995

respectably. $\rightsquigarrow$ respectably. A subtract-with-borrow generator fails the gap test when the maximum gap length exceeds the largest lag; see Vattulainen, Kankaala, Saarinen, and Ala-Nissila, *Computer Physics Communications* **86** (1995), 209–226, where a variety of other tests are also reported. Lagged Fibonacci generators, which are theoretically guaranteed to have equally distributed least-significant bits, still fail some simple variants of the 1-bit equidistribution test (see exercises 31 and 32).

Page 73 new paragraph before the exercises ________ 20 Aug 1995

As computers become faster, more random numbers are consumed than ever before, and random number generators that once were satisfactory are no longer good enough for sophisticated applications in physics, combinatorics, stochastic geometry, etc. George Marsaglia therefore introduced a number of *stringent tests*, which go well beyond classical methods like the gap and poker tests, in order to meet the new challenges. For example, he found that the sequence $X_{n+1} = (62605X_n + 113218009) \bmod 2^{29}$ had a noticeable bias in the following experiment: Generate 2^{21} random numbers X_n and extract their 10 leading bits $Y_n = \lfloor X_n/2^{19}\rfloor$. Count how many of the 2^{20} possible pairs (y, y') of 10-bit numbers do *not* occur among (Y_1, Y_2), (Y_2, Y_3), …, $(Y_{2^{21}-1}, Y_{2^{21}})$. There ought to be about 141909.33 missing pairs, with standard deviation ≈ 290.46 (see exercise 35). But six consecutive trials, starting with $X_1 = 1234567$, produced counts that were all between 1.5 and 3.5 standard deviations too low. The distribution was a bit too "flat" to be random — probably because 2^{21} numbers is a significant fraction, 1/256, of the entire period. A similar generator with multiplier 69069 and modulus 2^{30} proved to be better. Marsaglia and Zaman call this procedure a "monkey test," because it counts the number of two-character combinations that a monkey will miss after typing randomly on a keyboard with 1024 keys; they have analyzed several monkey tests in *Computers and Math.* **26**, 9 (Nov. 1993), 1–10.

▸**Page 75** line 4 ________ 02 Feb 1981

$\alpha X_k + \beta Y_k \rightsquigarrow \alpha U_k + \beta V_k$

Page 75 replacements for exercises 19 and 20 ______ 18 Jul 1995

19. [*M30*] (J. P. Butler.) Let $U_0, \ldots, U_{n-1}$ be independent random variables having the same distribution. Prove that the expected value of the serial correlation coefficient (23), averaged over all cases with nonzero denominator, is $-1/(n-1)$.

20. [*HM41*] Continuing the previous exercise, prove that the variance of (23) is equal to $n^2/(n-1)^2(n-2) - n^3\,\mathrm{E}\bigl((U_0-U_1)^4/D^2\bigr)/2(n-2)$, where D is the denominator of (23) and E denotes the expected value over all cases with $D \neq 0$. What is the asymptotic value of $\mathrm{E}\bigl((U_0-U_1)^4/D^2\bigr)$ when each U_j is uniformly distributed?

Page 75 new exercises ______ 20 Aug 1995

23. [*M22*] Let $\langle Y_n\rangle$ and $\langle Y'_n\rangle$ be integer sequences having period lengths λ and λ', respectively, with $0 \le Y_n, Y'_n < d$; also let $Z_n = (Y_n + Y'_{n+r}) \bmod d$, where r is chosen at random between 0 and $\lambda'-1$. Show that $\langle Z_n\rangle$ passes the t-dimensional serial test at least as well as $\langle Y_n\rangle$ does, in the following sense: Let $P(x_1,\ldots,x_t)$ and $Q(x_1,\ldots,x_t)$ be the probabilities that the t-tuple $(x_1,\ldots,x_t)$ occurs in $\langle Y_n\rangle$ and $\langle Z_n\rangle$:

$$P(x_1,\ldots,x_t) = \frac{1}{\lambda}\sum_{n=0}^{\lambda-1}\bigl[(Y_n,\ldots,Y_{n+t-1}) = (x_1,\ldots,x_t)\bigr];$$

$$Q(x_1,\ldots,x_t) = \frac{1}{\lambda\lambda'}\sum_{n=0}^{\lambda-1}\sum_{r=0}^{\lambda'-1}\bigl[(Z_n,\ldots,Z_{n+t-1}) = (x_1,\ldots,x_t)\bigr].$$

Then

$$\sum_{(x_1,\ldots,x_t)}\bigl(Q(x_1,\ldots,x_t) - d^{-t}\bigr)^2 \;\le\; \sum_{(x_1,\ldots,x_t)}\bigl(P(x_1,\ldots,x_t) - d^{-t}\bigr)^2.$$

24. [*HM35*] (G. Marsaglia.) Show that the serial test on n overlapping t-tuples $(Y_1,Y_2,\ldots,Y_t)$, $(Y_2,Y_3,\ldots,Y_{t+1})$, $\ldots$, $(Y_n,Y_1,\ldots,Y_{t-1})$ can be carried out as follows: For each string $\alpha = a_1\ldots a_m$ with $0 \le a_i < d$, let $N(\alpha)$ be the number of times α occurs as a substring of $Y_1Y_2\ldots Y_nY_1\ldots Y_{m-1}$, and let $P(\alpha) = P(a_1)\ldots P(a_m)$ be the probability that α occurs at any given position; individual digits may occur with differing probabilities $P(0)$, $P(1)$, $\ldots$, $P(d-1)$. Compute the statistic

$$V \;=\; \frac{1}{n}\sum_{|\alpha|=t}\frac{N(\alpha)}{P(\alpha)} \;-\; \frac{1}{n}\sum_{|\alpha|=t-1}\frac{N(\alpha)}{P(\alpha)}\,.$$

Then V should have the chi-square distribution with $d^t - d^{t-1}$ degrees of freedom when n is large. [*Hint:* Use exercise 3.3.1–25.]

Page 75 new exercise ______ 17 Jul 1995

25. [*M46*] Why is $C_1^{-1}C_2C_1^{-1} \approx -6C_1^{-1}$, when C_1 and C_2 are the matrices of (22)?

Page 75 new exercises ______ 06 Sep 1995

26. [*HM30*] Let U_1, U_2, $\ldots$, U_n be independent uniform deviates in $[0,1)$, and let $U_{(1)} \le U_{(2)} \le \cdots \le U_{(n)}$ be their values after sorting; also define the spacings $S_1 = U_{(2)} - U_{(1)}$, $\ldots$, $S_{n-1} = U_{(n)} - U_{(n-1)}$, $S_n = U_{(1)} + 1 - U_{(n)}$ and sorted spacings $S_{(1)} \le \cdots \le S_{(n)}$ as in the birthday spacings test. It is convenient in the following calculations to use the notation x_+^n as an abbreviation for the expression $x^n[x \ge 0]$.

(a) Given any real numbers s_1, s_2, $\ldots$, s_n, prove that the simultaneous inequalities $S_1 \ge s_1$, $S_2 \ge s_2$, $\ldots$, $S_n \ge s_n$ occur with probability $(1 - s_1 - s_2 - \cdots - s_n)_+^{n-1}$.

(b) Consequently the smallest spacing $S_{(1)}$ is $\le s$ with probability $1-(1-ns)_+^{n-1}$.
(c) What are the distribution functions $F_k(s) = \Pr(S_{(k)} \le s)$, for $1 \le k \le n$?
(d) Calculate the mean and variance of each $S_{(k)}$.

▸**27.** [*HM26*] (*Iterated spacings.*) In the notation of the previous exercise, show that the numbers $S'_1 = nS_{(1)}$, $S'_2 = (n-1)(S_{(2)} - S_{(1)})$, …, $S'_n = 1(S_{(n)} - S_{(n-1)})$ have the same joint probability distribution as the original spacings S_1, …, S_n of n uniform deviates. Therefore we can sort them into order, $S'_{(1)} \le \cdots \le S'_{(n)}$, and repeat this transformation to get yet another set of random spacings S''_1, …, S''_n, etc. Each successive set of spacings $S_1^{(k)}$, …, $S_n^{(k)}$ can be subjected to the Kolmogorov-Smirnov test, using

$$K^+_{n-1} = \sqrt{n-1}\max_{1\le j<n}\left(\frac{j}{n-1} - S_1^{(k)} - \cdots - S_j^{(k)}\right),$$

$$K^-_{n-1} = \sqrt{n-1}\max_{1\le j<n}\left(S_1^{(k)} + \cdots + S_j^{(k)} - \frac{j-1}{n-1}\right).$$

Examine the transformation from $(S_1, \ldots, S_n)$ to $(S'_1, \ldots, S'_n)$ in detail in the cases $n = 2$ and $n = 3$; explain why continued repetition of this process will break down eventually when it is applied to computer-generated numbers with finite precision. (One way to compare random number generators is to see how long they can continue to survive such a torture test.)

28. [*M26*] Let $b_{nrs}(m)$ be the number of n-tuples $(y_1, \ldots, y_n)$ with $0 \le y_j < m$ that have exactly r equal spacings and s zero spacings. Thus, the probability that $R = r$ in the birthday spacings test is $\sum_{s=0}^{r+1} b_{nrs}(m)/m^n$. Also let $p_n(m)$ be the number of partitions of m into at most n parts (exercise 5.1.1–15). (a) Express $b_{n00}(m)$ in terms of partitions. [*Hint:* Consider cases with small m and n.] (b) Show that there is a simple relation between $b_{nrs}(m)$ and $b_{(n-s)(r+1-s)0}(m)$ when $s > 0$. (c) Therefore there is an explicit formula for the probability that no spacings are equal.

29. [*M35*] Continuing exercise 28, find simple expressions for the generating functions $b_{nr}(z) = \sum_{m\ge 0} b_{nr0}(m)z^m/m$, when $r = 0$, 1, and 2.

30. [*HM41*] Continuing the previous exercises, prove that if $m = n^3/\alpha$ we have

$$p_n(m) = \frac{m^{n-1}e^{\alpha/4}}{n!\,(n-1)!}\left(1 - \frac{13\alpha^2}{288n} + \frac{169\alpha^4 + 2016\alpha^3 - 1728\alpha^2 - 41472\alpha}{165888n^2} + O(n^{-3})\right)$$

for fixed α as $n \to \infty$. Find a similar formula for $q_n(m)$, the number of partitions of m into n *distinct* positive parts. Deduce the asymptotic probabilities that the birthday spacings test finds R equal to 0, 1, and 2, to within $O(1/n)$.

Page 75 new exercises ________ 23 Sep 1995

▸**31.** [*M21*] The recurrence $Y_n = (Y_{n-24} + Y_{n-55}) \bmod 2$, which describes the least significant bits of the lagged Fibonacci generator 3.2.2–(7) as well as the second-least significant bits of 3.2.2–(7′), is known to have period length $2^{55}-1$; hence every possible nonzero pattern of bits $(Y_n, Y_{n+1}, \ldots, Y_{n+54})$ occurs equally often. Nevertheless, prove that if we generate 79 consecutive random bits Y_n, …, Y_{n+78} starting at a random point in the period, the probability is more than 51% that there are more 1s than 0s. If we use such bits to define a "random walk" that moves to the right when the bit is 1 and to the left when the bit is 0, we'll finish to the right of our starting point significantly more than half of the time. [*Hint:* Find the generating function $\sum_{k=0}^{79} \Pr(Y_n + \cdots + Y_{n+78} = k)\, z^k$.]

33. [*M20*] True or false: If X and Y are independent, identically distributed random variables with mean 0, and if they are more likely to be positive than negative, then $X+Y$ is more likely to be positive than negative.

34. [*HM32*] Find the asymptotic value of the probability that $k+l$ consecutive bits generated by the recurrence $Y_n = (Y_{n-l} + Y_{n-k}) \bmod 2$ have more 1s than 0s, when $k > 2l$ and the period length of this recurrence is $2^k - 1$, assuming that k is large.

Page 75 new exercise ——— 06 Jan 1997

▸**32.** [*HM30*] (J. H. Lindholm, 1968.) Suppose we generate random bits $\langle Y_n \rangle$ using the recurrence

$$Y_n = (a_1 Y_{n-1} + a_2 Y_{n-2} + \cdots + a_k Y_{n-k}) \bmod 2\,,$$

for some choice of $a_1, \ldots, a_k$ such that the period length is $2^k - 1$; start with $Y_0 = 1$ and $Y_1 = \cdots = Y_{k-1} = 0$. Let $Z_n = (-1)^{Y_n+1} = 2Y_n - 1$ be a random sign, and consider the statistic $S_m = Z_n + Z_{n+1} + \cdots + Z_{n+m-1}$, where n is a random point in the period.

a) Prove that $\mathrm{E}\, S_m = m/N$, where $N = 2^k - 1$.
b) What is $\mathrm{E}\, S_m^2$? Assume that $m \le N$. *Hint:* See exercise 3.2.2–16.
c) What would $\mathrm{E}\, S_m$ and $\mathrm{E}\, S_m^2$ be if the Z's were truly random?
d) Assuming that $m \le N$, prove that $\mathrm{E}\, S_m^3 = m^3/N - 6B(N+1)/N$, where

$$B = \sum_{0<i<j<m} \bigl[(Y_{i+1}Y_{i+2}\ldots Y_{i+k-1})_2 = (Y_{j+1}Y_{j+2}\ldots Y_{j+k-1})_2\bigr]\,(m-j)\,.$$

e) Evaluate B in the special case considered in the previous exercise: $m = 79$ and $Y_n = (Y_{n-24} + Y_{n-55}) \bmod 2$.

Page 75 new exercise ——— 01 Oct 1995

35. [*HM29*] Explain how to estimate the mean and variance of the number of two-letter combinations that do not occur consecutively in a random string of length n on an m-letter alphabet. Assume that m is large and $n \approx 2m^2$.

Page 77 replacement for (7) ——— 23 Oct 1989

$$\begin{aligned} ((x)) &= x - \lfloor x \rfloor - \tfrac12 + \tfrac12\delta(x) \\ &= x - \lceil x \rceil + \tfrac12 - \tfrac12\delta(x) \\ &= x - \bigl(\lfloor x \rfloor + \lceil x \rceil\bigr)/2. \end{aligned} \tag{7}$$

▸**Page 78** line 9 ——— 07 May 1997

$$\sum_{0\le x\le m_0} \rightsquigarrow \sum_{0\le x<m_0}$$

▸**Page 78** last line of Eq. (13) ——— 28 Jul 1981

$$\frac{b_0 x}{m} \rightsquigarrow \frac{b_0 x}{m_0}$$

Page 82 replacement for lines 3 and 4 above (35) ——— 18 Jul 1995

$$\frac{h}{k} + \frac{m_1 - m_3}{m_2} - \frac{m_2 - m_4}{m_3} + \frac{m_3 - m_5}{m_4} - \frac{m_4}{m_5} = \frac{h}{k} + a_1 - a_2 + a_3 - a_4.$$

▸**Page 85** replacement for (40) ______ 18 Jul 1995

$$-\frac{1}{2}\sum_{\substack{1\le j\le t\\ j\,\mathrm{odd}}} a_j - \sum_{\substack{1\le j\le t\\ j\,\mathrm{even}}} a_j \le \sigma(h,k,c) \le \sum_{\substack{1\le j\le t\\ j\,\mathrm{odd}}} a_j + \frac{1}{2}\sum_{\substack{1\le j\le t\\ j\,\mathrm{even}}} a_j - \frac{1}{2}. \qquad (40)$$

▸**Page 85** replacement for line 8 after (40) ______ 18 Dec 1982

$$-55 \le \sigma(a,m,c) \le 67.5,$$

▸**Page 92** line 9 ______ 08 Jul 1992

arrrangement $\rightsquigarrow$ arrangement

Page 94 in the third edition . . . . 20 Sep 1995

Eventually I want Volume 2 to include a thorough discussion of the LLL algorithm for short lattice vectors [Lenstra, Lenstra, and Lovász, Mathematische Annalen **261** *(1982), 515–534] and various subsequent improvements. I'm still uncertain about how to combine it with the material of Section 3.3.4, nor do I know where everything will fit best in the table of contents. (Already I apply their results in exercise 4.5.4–44.)*

Page 97 line −16 ______ 30 Oct 1995

"Schmidt orthogonalization $\rightsquigarrow$ "Gram-Schmidt orthogonalization

Page 98 improved kudology ______ 16 Apr 1995

line 11: a significant $\rightsquigarrow$ Gauss's significant
line 12: case. $\rightsquigarrow$ case (exercise 5).

▸**Page 98** first line of step S1 ______ 10 Aug 1995

Set $h \leftarrow a$ $\rightsquigarrow$ Set $t \leftarrow 2$, $h \leftarrow a$

Page 99 bottom two lines ______ 10 Aug 1995

greater than 1, . . . such phenomena $\rightsquigarrow$ greater than 1; but larger z's

Page 101 new paragraph to follow line 15 ______ 10 Aug 1995

The exhaustive search in steps S9–S11 reduces the value of s only rarely. One such case, found in 1982 by R. Carling and K. Levine, occurs when $a = 464680339$, $m = 2^{29}$, and $t = 5$; another case arose when the author calculated ν_6^2 for line 21 of Table 1, later in this section.

Page 102 replacement for the table entries ———————— 28 Aug 1995

Line	a	m	ν_2^2	ν_3^2	ν_4^2	ν_5^2	ν_6^2
1	23	10^8+1	530	530	530	530	447
2	2^7+1	2^{35}	16642	16642	16642	15602	252
3	$2^{18}+1$	2^{35}	34359738368	6	4	4	4
4	3141592653	2^{35}	2997222016	1026050	27822	1118	1118
5	137	256	274	30	14	6	4
6	3141592621	10^{10}	4577114792	1034718	62454	1776	542
7	3141592221	10^{10}	4293881050	276266	97450	3366	2382
8	4219755981	10^{10}	10721093248	2595578	49362	5868	820
9	4160984121	10^{10}	9183801602	4615650	16686	6840	1344
10	$2^{24}+2^{13}+5$	2^{35}	8364058	8364058	21476	16712	1496
11	5^{13}	2^{35}	33161885770	2925242	113374	13070	2256
12	$2^{16}+3$	2^{29}	536936458	118	116	116	116
13	1812433253	2^{32}	4326934538	1462856	15082	4866	906
14	1566083941	2^{32}	4659748970	2079590	44902	4652	662
15	69069	2^{32}	4243209856	2072544	52804	6990	242
16	1664525	2^{32}	4938916874	2322494	63712	4092	1038
17	314159269	$2^{31}-1$	1432232969	899290	36985	3427	1144
18	62089911	$2^{31}-1$	1977289717	1662317	48191	6101	1462
19	16807	$2^{31}-1$	282475250	408197	21682	4439	895
20	48271	$2^{31}-1$	1990735345	1433881	47418	4404	1402
21	40692	$2^{31}-249$	1655838865	1403422	42475	6507	1438
22	44485709377909	2^{46}	5.6×10^{13}	1180915002	1882426	279928	26230
23	31167285	2^{48}	3.2×10^{14}	4111841446	17341510	306326	59278
24	see (38)		2.4×10^{18}	4.7×10^{11}	1.9×10^{9}	3194548	1611610
25	see (39)		$(2^{31}-1)^2$	1.4×10^{12}	643578623	12930027	837632
26	see the text	2^{64}	8.8×10^{18}	6.4×10^{12}	4.1×10^{9}	45662836	1846368
27	see the text	$\approx 2^{78}$	$2^{62}+1$	4281084902	2.2×10^{9}	1.8×10^{9}	1862407
28	$2^{-24\cdot389}$	$\approx 2^{576}$	1.8×10^{173}	3.5×10^{115}	4.4×10^{86}	2×10^{69}	5×10^{57}
29	$(2^{32}-5)^{-400}$	$\approx 2^{1376}$	1.6×10^{414}	8.6×10^{275}	1×10^{207}	2×10^{165}	8×10^{137}

Page 102 new data requested . . . . 29 Aug 1995

Presumably people will soon publish additional interesting multipliers for the case $m = 2^{64}$ (or for prime m such as $2^{63}-25$ or $2^{64}-59$), as 64-bit computers become more common.

Page 102 with the displayed equations ———————— 29 Aug 1995

(37) $\rightsquigarrow$

Page 103 replacement for the table entries ——— 29 Aug 1995

$\lg\nu_2$	$\lg\nu_3$	$\lg\nu_4$	$\lg\nu_5$	$\lg\nu_6$	μ_2	μ_3	μ_4	μ_5	μ_6	Line
4.5	4.5	4.5	4.5	4.4	$2\epsilon^5$	$5\epsilon^4$	0.01	0.34	4.62	1
7.0	7.0	7.0	7.0	4.0	$2\epsilon^6$	$3\epsilon^4$	0.04	4.66	$2\epsilon^3$	2
17.5	1.3	1.0	1.0	1.0	3.14	$2\epsilon^9$	$2\epsilon^9$	$5\epsilon^9$	ϵ^8	3
15.7	10.0	7.4	5.1	5.1	0.27	0.13	0.11	0.01	0.21	4
4.0	2.5	1.9	1.3	1.0	3.36	2.69	3.78	1.81	1.29	5
16.0	10.0	8.0	5.4	4.5	1.44	0.44	1.92	0.07	0.08	6
16.0	9.0	8.3	5.9	5.6	1.35	0.06	4.69	0.35	6.98	7
16.7	10.7	7.8	6.3	4.8	3.37	1.75	1.20	1.39	0.28	8
16.5	11.1	7.0	6.4	5.2	2.89	4.15	0.14	2.04	1.25	9
11.5	11.5	7.2	7.0	5.3	$8\epsilon^4$	2.95	0.07	5.53	0.50	10
17.5	10.7	8.4	6.8	5.6	3.03	0.61	1.85	2.99	1.73	11
14.5	3.4	3.4	3.4	3.4	3.14	ϵ^5	ϵ^4	ϵ^3	0.02	12
16.0	10.2	6.9	6.1	4.9	3.16	1.73	0.26	2.02	0.89	13
16.1	10.5	7.7	6.1	4.7	3.41	2.92	2.32	1.81	0.35	14
16.0	10.5	7.8	6.4	4.0	3.10	2.91	3.20	5.01	0.02	15
16.1	10.6	8.0	6.0	5.0	3.61	3.45	4.66	1.31	1.35	16
15.2	9.9	7.6	5.9	5.1	2.10	1.66	3.14	1.69	3.60	17
15.4	10.3	7.8	6.3	5.3	2.89	4.18	5.34	7.13	7.52	18
14.0	9.3	7.2	6.1	4.9	0.41	0.51	1.08	3.22	1.73	19
15.4	10.2	7.8	6.1	5.2	2.91	3.35	5.17	3.15	6.63	20
15.3	10.2	7.7	6.3	5.2	2.42	3.24	4.15	8.37	7.16	21
22.8	15.1	10.4	9.0	7.3	2.48	2.42	0.25	3.10	1.33	22
24.1	16.0	12.0	9.1	7.9	3.60	3.92	5.27	0.97	3.82	23
30.5	19.4	15.4	10.8	10.3	1.65	0.29	3.88	0.02	4.69	24
31.0	20.2	15.6	11.8	9.8	3.14	1.49	0.44	0.69	0.66	25
31.5	21.3	16.0	12.7	10.4	1.50	3.68	4.52	4.02	1.76	26
31.0	16.0	15.5	15.4	10.4	$5\epsilon^5$	$4\epsilon^9$	$8\epsilon^5$	2.56	ϵ^4	27
288.	192.	144.	115.	95.9	2.27	3.46	3.92	2.49	2.98	28
688.	458.	344.	275.	229.	3.10	2.04	2.85	1.15	1.33	29
		upper bounds from (40):			3.63	5.92	9.87	14.89	23.87	

Page 104 lines 13–31 ——— 29 Aug 1995

exercise 11) ... a decade; $\leadsto$ exercise 11). Line 10 is interesting because it has high μ_3 in spite of very low μ_2 (see exercise 8).

Line 11 of Table 1 is a reminder of the good old days — it once was used extensively, following a suggestion of O. Taussky in the early 1950s. But computers for which 2^{35} was an appropriate modulus began to fade in importance during the late 60s, and they disappeared almost completely in the 80s, as machines with 32-bit arithmetic began to proliferate. This change to a comparatively small word size called for comparatively greater care. Line 12 was, alas, the generator actually used on such machines in most of the world's scientific computing centers for more than a decade;

Page 104 line −11 ——— 29 Aug 1995

(38) $\leadsto$ (37)

▸**Page 104** line −9 ——— 07 Oct 1981

$9X_n + 6X_{n+2} + X_{n+2} \leadsto 9X_n - 6X_{n+1} + X_{n+2}$

Page 104 bottom ______________________________ 29 Aug 1995

lines -5 to -3: Lines 23 and 24 ... line 30 $\leadsto$ Lines 13 and 14 are the Borosh-Niederreiter and Waterman multipliers for modulus 2^{32}. Lines 16 and 23 were found by M. Lavaux and F. Janssens in a computer search for spectrally good multipliers having a very high μ_2; line 22 is for the multiplier used with $c = 0$ and $m = 2^{48}$ in the Cray X-MP library; line 26

line -2: Line 25 was $\leadsto$ Line 15 was

Page 105 replacement for lines 1–3 ______________________________ 24 Aug 1995

for nearly cubical lattices in dimensions 2 through 5, partly because it is easy to remember [*Applications of Number Theory to Numerical Analysis* (Academic Press, 1972), 275].

Line 17 uses a random primitive root, modulo the prime $2^{31}-1$, as multiplier. Line 18 shows the spectrally best primitive root for $2^{31}-1$, found in an exhaustive search by G. S. Fishman and L. R. Moore III [*SIAM J. Sci. Stat. Comput.* **7** (1986), 24–45]. The adequate but less outstanding multiplier $16807 = 7^5$ in line 19 is actually used most often for that modulus, after being proposed by Lewis, Goodman, and Miller in *IBM Systems J.* **8** (1969), 136–146; for example, it has been one of the main generators in the popular IMSL subroutine library since 1971. The main reason for continued use of $a = 16807$ is that a^2 is less than the modulus m, hence $ax \bmod m$ can be implemented with reasonable efficiency in high-level languages using the technique of exercise 3.2.1.1–9. However, such small multipliers have known defects. S. K. Park and K. W. Miller noticed that the same technique applies also to certain multipliers greater than $\sqrt{m}$, so they asked G. S. Fishman to find the best "efficiently portable" multiplier in this wider class; the result appears in line 20 [*CACM* **31** (1988), 1192–1201]. Line 21 shows another good multiplier, due to P. L'Ecuyer [*CACM* **31** (1988), 742–749, 774]; this one uses a slightly smaller prime modulus.

When the generators of lines 20 and 21 are combined as suggested in Eq. 3.2.2–(15), so that the generated numbers $\langle Z_n \rangle$ satisfy

$$\begin{gathered} X_{n+1} = 48271X_n \bmod (2^{31}-1), \qquad Y_{n+1} = 40692Y_n \bmod (2^{31}-249), \\ Z_n = (X_n - Y_n) \bmod (2^{31}-1), \end{gathered} \tag{38}$$

exercise 32 shows that it is reasonable to rate $\langle Z_n \rangle$ with the spectral test for $m = (2^{31}-1)(2^{31}-249)$ and $a = 1431853894371298687$. (This value of a satisfies $a \bmod (2^{31}-1) = 48271$ and $a \bmod (2^{31}-249) = 40692$.) The results appear on line 24. We needn't worry too much about the low value of μ_5, since $\nu_5 > 1000$. Generator (38) has a period of length $(2^{31}-2)(2^{31}-250)/62 \approx 7 \times 10^{16}$.

Line 25 of the table represents the sequence

Page 105 new material to follow line 6 ______________________________ 19 Sep 1995

The last three lines of Table 1 are based on "add-with-carry" and "subtract-with-borrow" methods, which simulate linear congruential sequences that have extremely large moduli (see exercise 3.2.1.1–13). Line 27 is for the generator

$$X_n = (X_{n-1} + 65430X_{n-2} + C_n) \bmod 2^{31},$$
$$C_{n+1} = \lfloor (X_{n-1} + 65430X_{n-2} + C_n)/2^{31} \rfloor,$$

which corresponds to $\mathcal{X}_{n+1} = (65430 \cdot 2^{31} + 1)\mathcal{X}_n \bmod (65430 \cdot 2^{62} + 2^{31} - 1)$; the numbers in the table refer to the "super-values"

$$\mathcal{X}_n = (65430 \cdot 2^{31} + 1)X_{n-1} + 65430X_{n-2} + C_n$$

rather than to the values X_n actually computed and used as random numbers. Line 28 represents a more typical subtract-with-borrow generator

$$X_n = (X_{n-10} - X_{n-24} - C_n) \bmod 2^{24}, \quad C_{n+1} = [X_{n-10} < X_{n-24} + C_n],$$

but modified by generating 389 elements of the sequence and then using only the first (or last) 24. This generator, called RANLUX, was recommended by Martin Lüscher after it passed many stringent tests that previous generators failed [*Computer Physics Communications* **79** (1994), 100–110]. A similar sequence,

$$X_n = (X_{n-22} - X_{n-43} - C_n) \bmod (2^{32} - 5), \quad C_{n+1} = [X_{n-22} < X_{n-43} + C_n],$$

with 43 elements used after 400 are generated, appears in line 29; this sequence is discussed in the answer to exercise 3.2.1.2–22. In both cases the data in the Table refers to the spectral test on multiprecision numbers $\mathcal{X}_n$ instead of the individual "digits" X_n, but the high μ values indicate that the process of generating 389 or 400 numbers before selecting 24 or 43 is an excellent way to remove biases due to the extreme simplicity of the generation scheme.

Page 105 replacement for lines 10–12 ———— 17 Apr 1995

$$\nu_t \le \gamma_t^{1/2} m^{1/t}, \tag{40}$$

where γ_t takes the respective values

$$(4/3)^{1/2}, \quad 2^{1/3}, \quad 2^{1/2}, \quad 2^{3/5}, \quad (64/3)^{1/6}, \quad 4^{3/7}, \quad 2 \tag{41}$$

line 14: 332. ⇝ 332; J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups* (New York: Springer, 1988), p. 20.

▸**Page 107** line 8 ———— 03 Apr 1985

conguential ⇝ congruential

▸**Page 108** line 2 before (51) ———— 01 May 1997

3.2.1–21 ⇝ 3.2.1.2–21

Page 109 line −2 ———— 29 Aug 1995

lines 8, 19, and 23 ⇝ lines 8 and 13

Page 110 line 26 ———— 19 Sep 1995

Dieter. ⇝ Dieter. [See *Math. Comp.* **29** (1975), 827–833.]

Page 110 line -12 ——— 02 Aug 1995

(1969), 70–112 $\rightsquigarrow$ (Philadelphia: SIAM, 1969), 70–111

Page 110 lines -8 through -3 ——— 19 Sep 1995

Harald Niederreiter ... 1117–1132; $\rightsquigarrow$ R. G. Stoneham showed, by using estimates of exponential sums, that $p^{1/2+\epsilon}$ or more elements of the sequence $a^k X_0 \bmod p$ have asymptotically small discrepancy, when a is a primitive root modulo the prime p [*Acta Arithmetica* **22** (1973), 371–389]. This work was extended as explained above in a number of papers by Harald Niederreiter [*Math. Comp.* **28** (1974), 1117–1132; **30** (1976), 571–597; *Advances in Math.* **26** (1977), 99–181; *Bull. Amer. Math. Soc.* **84** (1978), 957–1041]. See also Niederreiter's book *Random Number Generation and Quasi-Monte Carlo Methods* (Philadelphia: SIAM, 1992).

▸**Page 111** line 4 of exercise 2 ——— 09 May 1997

$f(x_1,\ldots,x_t) \rightsquigarrow f(x_1,\ldots,x_t)^{1/2}$

Page 112 exercise 20 ——— 11 Aug 1995

line 1: [*M21*] $\rightsquigarrow$ [*M23*]
line 2: $a \bmod 8 = 5.$ $\rightsquigarrow$ $a \bmod 8 = 3$ or 5. (See exercise 3.2.1.2–9.)

▸**Page 112** lines 5 and 6 of exercise 23 ——— 09 May 1997

$n \rightsquigarrow t$ (in 4 places)

Page 112 bottom two lines ——— 21 Mar 1995

[*Matematicheskie Zametki*, to appear] $\rightsquigarrow$

▸**Page 113** line 1 of exercise 27 ——— 30 Jul 1996

$r(u_1,\ldots,u_m) \rightsquigarrow r(u_1,\ldots,u_t)$

▸**Page 113** line 4 of exercise 28 ——— 09 May 1997

$\phi(m-1) \rightsquigarrow \varphi(m-1)$

▸**Page 113** exercise 30 ——— 03 Apr 1985

$r_{\max} \le m/\max(a_1,\ldots,a_s) \rightsquigarrow r_{\max} = O\bigl(\max(a_1,\ldots,a_s)/m\bigr)$

Page 113 new exercise ——— 29 Aug 1995

▸**32.** [*M21*] Let $m_1 = 2^{31}-1$ and $m_2 = 2^{31}-249$ be the moduli of generator (38). (a) Show that if $U_n = (X_n/m_1 - Y_n/m_2) \bmod 1$, we have $U_n \approx Z_n/m_1$. (b) Let $W_0 = (X_0 m_2 - Y_0 m_1) \bmod m$ and $W_{n+1} = aW_n \bmod m$, where a and m have the values stated in the text following (38). Prove that there is a simple relation between W_n and U_n.

Page 115 five lines before (3) ——— 02 Mar 1981

(1976) $\rightsquigarrow$ (1977)

▸**Page 116** line 1 ——— 16 Apr 1995

$x_j = j \rightsquigarrow x_{j+1} = j$

Page 116 new paragraph to follow line 7 ——— 15 Apr 1996

The probabilities p_j can be represented implicitly by nonnegative weights $w_1, w_2, \ldots, w_k$; if we denote the sum of the weights by W, then $p_j = w_j/W$. In many applications the individual weights vary dynamically. Matias, Vitter, and Ni [*SODA* **4** (1993), 361–370] have shown how to update a weight and generate X in constant expected time.

Page 117 five lines after (10) ——— 02 Mar 1981

28 (1958), 610 ⇝ **29** (1958), 610–611

▸**Page 118** line 12 ——— 19 Feb 1997

derivative ⇝ differential

Page 118 replacement for (12) ——— 20 Aug 1995

$$F(x) = \operatorname{erf}(x/\sqrt{2}\,) = \sqrt{\frac{2}{\pi}}\int_0^x e^{-t^2/2}\,dt, \qquad x \geq 0, \tag{12}$$

▸**Page 125** line −2 ——— 28 Apr 1981

$-4/\ln U$ ⇝ $-4\ln U$

Page 127 line −12 ——— 20 Sep 1996

and F. ⇝ and F. An interesting alternative that requires fewer auxiliary table entries has been suggested by Ahrens and Dieter in *CACM* **31** (1988), 1330–1337.

Page 128 line 1 after (32) ——— 25 Jun 1982

must be avoided. ⇝ must be treated specially; we can substitute any convenient value ϵ for 0, since the probability of this case is extremely small.

▸**Page 128** corrections to Algorithm S ——— 21 Aug 1995

lines 2 and 6: t-bit ⇝ $(t+1)$-bit
line 5: 2^{1-t} ⇝ 2^{-t}
line 7: $(.b_1b_2\ldots b_t)_2$ ⇝ $(.b_0b_1\ldots b_t)_2$
line 7: leading j bits ⇝ leading $j+1$ bits
line 8: value of j ⇝ number of discarded bits

Page 129 line 16 ——— 25 Jun 1982

deviates. ⇝ deviates. Moreover, there is a substantial risk that the product $U_1\ldots U_{\lfloor a\rfloor}$ will cause floating-point underflow.

Page 129 line 18 ——— 16 Apr 1995

subroutines. ⇝ subroutines. [Cf. *Ann. Inst. Stat. Math.* **13** (1962), 231–237.]

Page 129 the lines following Algorithm A ——— 01 Mar 1982

line 1: [do not indent this line]
lines 2–3: [omit these lines]
insert new paragraph after line 7: For a somewhat complicated but significantly faster algorithm, which generates a gamma deviate in about twice the time

to generate a normal deviate, see J. H. Ahrens and U. Dieter, *CACM* **25** (1982), 47–54. This article contains an instructive discussion of the design principles used to construct the algorithm.

Page 129 three lines before (35) 04 May 1981

see G. Marsaglia, $\rightsquigarrow$ see E. B. Wilson and M. M. Hilferty, *Proc. Nat. Acad. Sci.* **17** (1931), 684–688; G. Marsaglia,

▸**Page 130** line −15 02 Feb 1981

$Y_1/\sqrt{(1-2\nu)(1-z)} \rightsquigarrow Y_1/\sqrt{(1-2\nu)(1-Z)}$

Page 133 lines 7 and 8 24 Aug 1995

The forthcoming book . . . Dieter $\rightsquigarrow$ A facsimile of a letter from von Neumann dated May 21, 1947, in which the rejection method first saw the light of day, appears in *Stanislaw Ulam 1909–1984*, a special issue of *Los Alamos Science* (Los Alamos National Lab., 1987), 135–136. The book *Non-Uniform Random Variate Generation* by L. Devroye (Springer, 1986)

Page 133 new paragraph to follow line 10 05 Oct 1995

W. Hörmann and G. Derflinger [*ACM Trans. Math. Software* **19** (1993), 489–495] have pointed out that it can be dangerous to use the rejection method in connection with linear congruential generators that have small multipliers $a \approx \sqrt{m}$.

▸**Page 136** line 19 04 May 1981

(J. L. Bentley and J. D. Saxe.) Fint $\rightsquigarrow$ (J. L. Bentley and J. B. Saxe.) Find

Page 136 new exercise 24 Sep 1981

30. [*M30*] Explain how to generate a set of random points (X_j, Y_j) such that, if R is any rectangle of area α contained in the unit square, the number of (X_j, Y_j) lying in R has the Poisson distribution with mean $\alpha\mu$.

Page 136 new exercise 24 Aug 1995

31. [*20*] Algorithms P, M, F, and R generate normal deviates by consuming an unknown number of uniform random variables U_1, U_2, ... How can they be modified so that the output is a function of just one U?

Page 139 line 5 21 Aug 1995

all times $\rightsquigarrow$ all times, and the algorithm becomes much simpler

Page 139 new paragraph to follow line 14 22 Aug 1995

Significant improvements to both Algorithms S and R are possible, when n/N is small, if we generate a single random variable to tell us how many records should be skipped instead of deciding whether or not to skip each record. (See exercise 8.)

Page 140 lines 1–5 ——— 21 Aug 1995

This algorithm ... For $\rightsquigarrow$ This algorithm was first published by R. A. Fisher and F. Yates [*Statistical Tables* (London, 1938), Example 12], in ordinary language, and by R. Durstenfeld [*CACM* **7** (1964), 420] in computer language. If we merely wish to generate a random permutation of $\{1, \ldots, t\}$ instead of shuffling a given sequence $(X_1, \ldots, X_t)$, we can avoid the exchange operation $X_k \leftrightarrow X_j$ by letting j increase from 1 to t and setting $X_j \leftarrow X_k$, $X_k \leftarrow j$; see D. E. Knuth, *The Stanford GraphBase* (ACM Press, 1993), 104.

R. Salfi [*COMPSTAT 1974* (Vienna, 1974), 28–35] pointed out that Algorithm P cannot possibly generate more than m distinct permutations when we obtain the uniform U's with a linear congruential sequence of modulus m, or indeed whenever we use a recurrence $U_{n+1} = f(U_n)$ for which U_n can take only m different values, because the final permutation in such cases is entirely determined by the value of the first U that is generated. Thus, for example, if $m = 2^{32}$, certain permutations of 13 elements will never occur, since $13! \approx 1.45 \times 2^{32}$. We don't really *want* to see all 13! permutations, usually; yet it is disconcerting to know that the excluded ones are determined by a fairly simple mathematical rule such as a lattice structure (see Section 3.3.4).

This problem does not arise when we use a lagged Fibonacci generator like 3.2.2–(7) with a sufficiently long period. But even with such methods we cannot get all permutations uniformly unless we are able to specify at least $t!$ different seed values to initialize the generator. In other words, we can't get $\lg t!$ truly random bits out unless we put $\lg t!$ truly random bits in. Section 3.5 shows that we need not despair about this.

Algorithm S can easily be modified to yield a random permutation of a random combination (see exercise 14). For

Page 140 replacement for exercise 8 ——— 22 Aug 1995

▸**8.** [*M30*] (J. S. Vitter.) Algorithm S computes one uniform deviate for each input record it handles. The purpose of this exercise is to consider a more efficient approach in which we calculate more quickly the proper number X of input records to skip before the first selection is made.

(a) What is the probability that $X \geq k$, given k?

(b) Show that the result of (a) allows us to calculate X by generating only one uniform U and then doing $O(X)$ other calculations.

(c) Show that we may also set $X \leftarrow \min(Y_N, Y_{N-1}, \ldots, Y_{N-n+1})$, where the Y's are independent and each Y_t is a random integer in the range $0 \leq Y_t < t$.

(d) For maximum speed, show that X can also be calculated in $O(1)$ steps, on the average, using a "squeeze method" like Eq. 3.4.1-(18).

▸**Page 141** exercise 16 ——— 22 Feb 1983

Delete this exercise.

Page 142 line 1 ——— 18 Jan 1981

3.5. $\rightsquigarrow$ ***3.5.**

Page 143 line 16 ———— 02 Feb 1981

$U_1, U_2, \leadsto U_0, U_1,$

Page 144 line 14 and throughout the books .... 25 Mar 1995

$[0,1) \leadsto [0\,..\,1)$

Page 154 bottom two lines ———— 22 Aug 1995

It is quite likely ... will satisfy $\leadsto$ The explicit-looking sequence $\langle \theta^n \bmod 1\rangle$ actually does, however, satisfy

Page 155 lines 1–7 ———— 22 Aug 1995

The following ... $\lfloor n \log n\rfloor$. $\leadsto$ J. F. Koksma proved that $\langle \theta^{s_n} \bmod 1\rangle$ is 1-distributed for almost all $\theta > 1$, if $\langle s_n\rangle$ is any sequence of distinct positive integers [*Compositio Math.* **2** (1935), 250–258]; H. Niederreiter and R. F. Tichy extended Koksma's theorem to prove that $\langle \theta^{s_n}\rangle$ is in fact ∞-distributed for almost all $\theta > 1$ [*Mathematika* **32** (1985), 26–32]. Only countably many sequences $\langle s_n\rangle$ are effectively definable, so $\langle \theta^n \bmod 1\rangle$ almost always satisfies R4.

Page 157 line 22 ———— 18 Mar 1995

L. H. Ramshaw ... to appear. $\leadsto$ S. Haber, *J. Research Nat. Bur. Standards* **B70** (1966), 127–136; R. Béjian and H. Faure, *Comptes Rendus* **A285** (Paris, 1977), 313–316; H. Faure, *J. Number Theory* **22** (1986), 4–20; S. Tezuka, *ACM Trans. Modeling and Comput. Sim.* **3** (1993), 99–107. L. H. Ramshaw has shown that the sequence $\langle \phi n \bmod 1\rangle$ is slightly more equally distributed than $\langle V_n\rangle$; see *J. Number Theory* **13** (1981), 138–175.

Page 157 line −5 ———— 28 Jul 1981

$[B[a_1,\ldots,a_r]$ full?$]$ $\leadsto$ [Is case $[a_1,\ldots,a_r]$ unfinished?]

▸**Page 158** line 27 ———— 28 Jul 1981

$s_1 < s_2 \le \cdots \leadsto s_1 < s_2 < \cdots$

Page 160 line 24 through page 161 line 8 ———— 16 May 1996

distributed. The next ... $\leadsto$ distributed. For $0 < \epsilon < 1$, let $B(r,\epsilon)$ ... with $|\nu(r) - \frac{1}{2}r| \ge 1 + \epsilon r$. Exercise 1.2.10–22 proves that $C(r,\epsilon) \le 2^{r+1}e^{-\epsilon^2 r}$; hence by (32),

$$B(r,\epsilon) \text{ has measure} \le 2^{-r}C(r,\epsilon) < 2e^{-\epsilon^2 r}. \qquad (33)$$

▸**Page 162** replacement for the lines following Definition Q1 ———— 15 May 1997

According to this definition, for example, there are 178 nonrandom binary sequences of length 11:

00000001111 10000000111 11000000011 11100000001 11110000000
00000001110 10000000110 11000000010 11100000000 11010000000
00000001101 10000000101 11000000001 10100000001 10110000000
00000001011 10000000011 01000000011 01100000001 01110000000
00000000111

▸**Page 163** line 3 before (35) ———— 20 Nov 1981

Kolomogorov $\leadsto$ Kolmogorov

Page 163 line −3 ________ 14 Sep 1996

21. $\leadsto$ 21. A generalized spectral test, based on discrete Fourier transforms, can be used to test how well a sequence measures up to Definition Q1 [cf. A. Compagner, *Physical Review* **E52** (1995), 5634–5645].

Page 164 line 4 ________ 02 Feb 1981

defined in exercise 1.1–8 $\leadsto$ discussed in Section 1.1

Page 164 lines 20 and 21 ________ 22 Apr 1996

Levin, *Doklady Akad. Nauk SSSR* **212** (1973), 548–550.

F. Summary, history, and bibliography.

$\leadsto$

Levin, *Doklady Akad. Nauk SSSR* **212** (1973), 548–550; *Information and Control* **61** (1984), 15–37.

F. Pseudorandom numbers. It is comforting from a theoretical standpoint to know that random finite sequences of various flavors exist, but such theorems don't answer the questions faced by real-world programmers. More recent developments have led to a more relevant theory, based on the study of *sets* of finite sequences. More precisely, we consider *multisets* in which sequences may appear more than once.

Let S be a multiset containing bit strings (i.e., binary sequences) of length N; we call S an N-*source*. Let $\$_N$ denote the special N-source that contains all 2^N possible N-bit strings. Each element of S represents a sequence that we might use as a source of pseudorandom bits; choosing different "seed" values leads to different elements of S. For example, S might be $\{B_1B_2\ldots B_N \mid B_j$ is the most significant bit of $X_j\}$ in the linear congruential sequence defined by $X_{j+1} = (aX_j + c) \bmod 2^e$, where there is one string $B_1B_2\ldots B_N$ for each of the 2^e starting values X_0.

The basic idea of pseudorandom sequences, as we have seen throughout this chapter, is to get N bits that appear to be random, although we rely only on a few "truly random" bits when we choose the seed value. In the example just considered, we need e truly random bits to select X_0; in general, selecting a member of S amounts to using $\lg|S|$ truly random bits, after which we proceed deterministically. If $N = 10^6$ and $|S| = 2^{32}$, we are getting more than 30,000 "apparently random" bits for each truly random bit expended. With $\$_N$ instead of S, we get no such amplification, because $\lg|\$_N| = N$.

What does it mean to be "apparently random"? A. C. Yao proposed a good definition in 1982: Consider any algorithm A that looks at a bit string $B = B_1\ldots B_N$ and outputs $A(B) = 0$ or 1. We may think of A as a test for randomness; for example, A might compute the distribution of runs of consecutive 0s and 1s, outputting 1 if the run lengths differ significantly from the expected distribution. Whatever A does, we can consider the probability $P(A, S)$ that $A(B) = 1$ when B is a randomly chosen element of S, and we can compare it to the probability $P(A, \$_N)$ that $A(B) = 1$ when B is a truly random bit string of length N. If $P(A,S)$ is extremely close to $P(A,\$_N)$ for all statistical tests A, we cannot tell the difference between the sequences of S and truly random binary sequences.

Definition P. *We say that an N-source S passes statistical test A with tolerance ϵ if $\bigl|P(A,S)-P(A,\$_N)\bigr|<\epsilon$. It fails the test if $\bigl|P(A,S)-P(A,\$_N)\bigr|\ge\epsilon$.*

The algorithm A need not be designed by statisticians. *Any* algorithm can be considered a statistical test for randomness, according to Definition P. We also allow A to flip coins (i.e., to use truly random bits) as it performs its calculations. The only requirement is that A must output 0 or 1.

Well, actually there is another requirement: We insist that A must deliver its output in a reasonable time, at least on the average. We're not interested in algorithms that will take years to run, because we will never notice any disparities between S and $\$_N$ if our computers cannot detect them during our lifetime. The sequences of S contain only $\lg|S|$ bits of information, so there surely are algorithms that will eventually detect the redundancy; but we don't care, as long as S is able to pass all the tests that really matter.

These qualitative ideas can be quantified, as we will now see. In the following discussion, the *running time* $T(A)$ of an algorithm A on N-bit strings is defined to be the maximum of the expected number of steps needed to output $A(B)$, maximized over all $B\in\$_N$; the expected number is averaged over all coin flips made by the algorithm.

The first step in our quantitative analysis is to show that we may restrict the tests to be of a very special kind. Let A_k be an algorithm that depends only on the first k bits of the input string $B=B_1\ldots B_N$, where $0\le k<N$, and let $A_k^P(B)=\bigl(A_k(B)+B_{k+1}+1\bigr)\bmod 2$. Thus A_k^P outputs 1 if and only if A_k has successfully predicted B_{k+1}; we call A_k^P a *prediction* test.

Lemma P1. *Let S be an N-source. If S fails test A with tolerance ϵ, there is a $k\in\{0,1,\ldots,N-1\}$ and a prediction test A_k^P with $T(A_k^P)\le T(A)+O(N)$ such that S fails A_k^P with tolerance ϵ/N.*

Proof. By complementing the output of A, if necessary, we may assume that $P(A,S)-P(A,\$_N)\ge\epsilon$. Consider the algorithms F_k that begin by flipping $N-k$ coins and replacing $B_{k+1}\ldots B_N$ by random bits $B'_{k+1}\ldots B'_N$ before executing A. Algorithm F_N is the same as A, while F_0 acts on S as if A were acting on $\$_N$. Let $p_k=P(F_k,S)$. Since $\sum_{k=0}^{N-1}(p_{k+1}-p_k)=p_N-p_0=P(A,S)-P(A,\$_N)\ge\epsilon$, there is some k such that $p_{k+1}-p_k\ge\epsilon/N$.

Let A_k^P be the algorithm that performs the computations of F_k and predicts the value $\bigl(F_k(B)+B'_{k+1}+1\bigr)\bmod 2$; in other words, it outputs $A_k^P(B)=\bigl(F_k(B)+B_{k+1}+B'_{k+1}\bigr)\bmod 2$. A careful analysis of probabilities shows that $P(A_k^P,S)-P(A_k^P,\$_N)=p_{k+1}-p_k$. (See exercise 40.) ▌

Most N-sources S of practical interest are *shift-symmetric* in the sense that every substring $B_1\ldots B_k$, $B_2\ldots B_{k+1}$, $\ldots$, $B_{N-k+1}\ldots B_N$ of length k has the same probability distribution. This holds, for example, when S corresponds to a linear congruential sequence as explained above. In such cases we can improve on Lemma P1 by taking $k = N - 1$:

Lemma P2. *If S is a shift-symmetric N-source that fails test A with tolerance ϵ, there is an algorithm A' with $T(A') \le T(A) + O(N)$ that predicts B_N from $B_1 \ldots B_{N-1}$ with probability at least $\frac{1}{2} + \epsilon/N$.*

Proof. If $P(A,S) > P(A,\$_N)$, let A' be the A_k^P in the proof of Lemma P1, but applied to $B_{N-k} \ldots B_{N-1}0 \ldots 0$ instead of $B_1 \ldots B_N$. Then A' has the same average behavior, because of shift-symmetry. If $P(A,S) < P(A,\$_N$, let A' be $1 - A_k^P$ in the same fashion. Clearly $P(A',\$_N) = \frac{1}{2}$. ▮

Now let's specialize S even more, by supposing that each of the sequences $B_1 B_2 \ldots B_N$ has the form $f\bigl(g(X_0)\bigr) f\bigl(g^2(X_0)\bigr) \ldots f\bigl(g^N(X_0)\bigr)$ as X_0 ranges over some set X, where g is a permutation of X and $f(x)$ is 0 or 1 for all $x \in X$. Our linear congruential example satisfies this restriction, with $X = \{0, 1, \ldots, 2^e - 1\}$, $g(x) = (ax + c) \bmod 2^e$, and $f(x) =$ most significant bit of x. Such N-sources will be called *iterative*.

Lemma P3. *If S is an iterative N-source that fails test A with tolerance ϵ, there is an algorithm A' with $T(A') \le T(A) + O(N)$ that predicts B_1 from $B_2 \ldots B_N$ with probability at least $\frac{1}{2} + \epsilon/N$.*

Proof. An iterative N-source is shift-symmetric, and so is its reflection $S^R = \{B_N \ldots B_1 \mid B_1 \ldots B_N \in S\}$. Therefore Lemma P2 applies to S^R. ▮

The permutation $g(x) = (ax + c) \bmod 2^e$ is easy to invert, in the sense that we can determine x from $g(x)$ whenever a is odd. But many easily computed permutation functions are "one-way" — hard to invert — and we will see that this makes them provably good sources of pseudorandom numbers.

Lemma P4. *Let S be an iterative N-source corresponding to f, g, and X. If S fails test A with tolerance ϵ, there is an algorithm G that correctly guesses $f(x)$, given $g(x)$, with probability $\ge \frac{1}{2} + \epsilon/N$, when x is a random element of X. The running time $T(G)$ is at most $T(A) + O(N)\bigl(T(f) + T(g)\bigr)$.*

Proof. Given $y = g(x)$, the desired algorithm G computes $B_2 = f\bigl(g(x)\bigr)$, $B_3 = f\bigl(g^2(x)\bigr)$, ..., $B_N = f\bigl(g^{N-1}(x)\bigr)$ and applies the algorithm A' of Lemma P3. It guesses $f(x) = B_1$ with probability $\ge \frac{1}{2} + \epsilon/N$, because $B_1 \ldots B_N$ is the element of S that corresponds to $X_0 = g^{-1}(x)$ and g is a permutation of X. ▮

In order to use Lemma P4, we need to amplify the ability to guess a single bit $f(x)$ to an ability to guess x itself, given only the value of $g(x)$. There is a nice general way to do this, using the properties of Boolean functions, if we extend S so that many different functions $f(x)$ need to be guessed. (However, the method is somewhat technical, so the first-time reader may want to skip down to Theorem G before looking closely at the details that follow.)

Suppose $G(z_1 \ldots z_R)$ is a binary-valued function on R-bit strings that is good at guessing a function of the form $f(z_1 \ldots z_R) = (x_1 z_1 + \cdots + x_R z_R) \bmod 2$

for some fixed $x = x_1 \ldots x_R$. It is convenient to measure the success of G by computing the expected value

$$s = \mathrm{E}\bigl((-1)^{G(z_1 \ldots z_R)+x_1z_1+\cdots+x_Rz_R}\bigr)\,,$$

averaged over all possibilities for $z_1 \ldots z_R$. This is the sum of correct guesses minus incorrect guesses, divided by 2^R; so if p is the probability that G is correct, we have $s = p - (1-p)$, or $p = \frac{1}{2} + \frac{1}{2}s$.

For example, suppose $R = 4$ and $G(z_1z_2z_3z_4) = [z_1 \neq z_2][z_3 + z_4 < 2]$. This function has success rate $s = \frac{3}{4}$ (and $p = \frac{7}{8}$) if $x = 1100$, because it equals $x \cdot z \bmod 2 = (z_1 + z_2) \bmod 2$ for all 4-bit strings z except 0111 or 1011. It also has success rate $\frac{1}{4}$ when $x = 0000, 0001, 1101$, or 1110; so there are five plausible possibilities for x. The other eleven x's make $s \le 0$.

The following algorithm magically discovers x in most cases when G is a successful guesser in the sense just described. More precisely, it constructs a short list that has a good chance of containing x.

Algorithm L. *Given a binary-valued function $G(z_1 \ldots z_R)$ and a positive integer k, this algorithm outputs a list of 2^k binary sequences $x = x_1 \ldots x_R$ with the property that x is likely to be output when $G(z_1 \ldots z_R)$ is a good approximation to $(x_1z_1 + \cdots + x_Rz_R) \bmod 2$.*

L1. [Construct a random matrix.] Generate random bits B_{ij} for $1 \le i \le k$ and $1 \le j \le R$.

L2. [Compute signs.] For $1 \le i \le R$, and for all bit strings $b = b_1 \ldots b_k$, compute

$$h_i(b) = \sum_{c \neq 0} (-1)^{b \cdot c + G(cB + e_i)}$$

where e_i is the R-bit string $0\ldots010\ldots0$ having 1 in position i, and where cB is the string $d_1 \ldots d_R$ with $d_j = (B_{1j}c_1 + \cdots + B_{kj}c_k) \bmod 2$. (In other words the binary vector $c_1 \ldots c_k$ is multiplied by the $k \times R$ binary matrix B.) The sum is taken over all $2^k - 1$ values $c_1 \ldots c_k \neq 0 \ldots 0$. It can be evaluated for each i with $k \cdot 2^k$ additions and subtractions, using Yates's method for the Walsh transform; see the remarks following Eq. 4.6.4–(38).

L3. [Output the guesses.] For all 2^k choices of $b = b_1 \ldots b_k$, output the string $x(b) = [h_1(b) < 0] \ldots [h_R(b) < 0]$. ▮

To prove that Algorithm L works properly, we must show that a given string x will probably be output whenever it deserves to be. Notice first that if we change G to G', where $G'(z) = (G(z) + z_j) \bmod 2$, the original $G(z)$ is a good approximation to $x \cdot z \bmod 2$ if and only if the new $G'(z)$ is a good approximation to $(x + e_j) \cdot z \bmod 2$, where e_j is the unit-vector string defined in step L2. Moreover, if we apply the algorithm to G' instead of G, we get

$$h'_i(b) = \sum_{c \neq 0} (-1)^{b \cdot c + G(cB + e_i) + (cB + e_i) \cdot e_j} = (-1)^{\delta_{ij}} h_i\bigl((b + B_j) \bmod 2\bigr)\,,$$

where B_j is column j of B. Therefore step L3 outputs the vectors $x'(b) = x\bigl((b+B_j) \bmod 2\bigr) + e_j$, modulo 2. As b runs through all k-bit strings, so does $(b+B_j) \bmod 2$, and the effect is to complement bit j of every x in the output.

We need therefore prove only that the vector $x = 0\ldots0$ is likely to be output whenever $G(z)$ is a good approximation to the constant function 0. We will show, in fact, that $x(0\ldots0)$ equals $0\ldots0$ in step L3 with high probability, whenever $G(z)$ is a lot more likely to be 0 than 1 and k is sufficiently large. More precisely, the condition

$$\sum_{c\neq 0}(-1)^{G(cB+e_i)} > 0$$

holds for $1 \le i \le R$ with probability $> \frac{1}{2}$, if $s = \mathrm{E}\bigl((-1)^{G(z)}\bigr)$ is positive when averaged over all 2^R possibilities for z and if k is large enough.

The key observation is that, for each fixed $c = c_1\ldots c_k \neq 0\ldots0$, the string $d = cB$ is uniformly distributed: Every value of d occurs with probability $1/2^R$, because the bits of B are random. Furthermore, when $c \neq c' = c'_1\ldots c'_k$, the strings $d = cB$ and $d' = c'B$ are *independent*: Every value of the pair (d,d') occurs with probability $1/2^{2R}$. Therefore we can argue as in the proof of Chebyshev's inequality that $\sum_{c\neq0}(-1)^{G(cB+e_i)}$ is negative for any fixed i with probability at most $1/\bigl((2^k-1)s^2\bigr)$. (Exercise 42 contains the details.) It follows that $R/\bigl((2^k-1)s^2\bigr)$ is an upper bound on the probability that $x(0)$ is nonzero in step L3.

Theorem G. *If $s = \mathrm{E}\bigl((-1)^{G(z)+x\cdot z}\bigr) > 0$ and $2^k > \lceil 2R/s^2\rceil$, Algorithm L outputs x with probability $\ge \frac{1}{2}$. The running time is $O(k2^kR)$ plus the time to make 2^kR evaluations of G.* ∎

Now we are ready to prove that the muddle-square sequence of Eq. 3.2.2–(17) is a good source of (pseudo)random numbers. Suppose $2^{R-1} < M = PQ < 2^R$, where P and Q are prime numbers of the form $4k+3$ in the respective ranges $2^{(R-2)/2} < P < 2^{(R-1)/2}$, $2^{R/2} < Q < 2^{(R+1)/2}$. We will call M an R-bit *Blum integer*, because the importance of such numbers for cryptography was first pointed out by Manuel Blum [*COMPCON* **24** (Spring 1982), 133–137]. Blum originally suggested that P and Q both have $R/2$ bits, but Algorithm 4.5.4D shows that it is better to choose P and Q as stated here so that $Q-P > .29\times 2^{R/2}$.

Choose X_0 at random in the range $0 < X_0 < M$, with $X_0 \perp M$; also let Z be a random R-bit mask. We can construct an iterative N-source S by letting X be the set of all (x,z,m) that are possibilities for (X_0,Z,M), with the further restriction that $x \equiv a^2$ (modulo m) for some a. The function $g(x,z,m) = (x^2 \bmod m, z, m)$ is easily shown to be a permutation on X (see, for example, exercise 4.5.4–35). The function $f(x,z,m)$ that extracts bits in this iterative source is $x\cdot z \bmod 2$. Our starting value (X_0,Z,M) isn't necessarily in X, but $g(X_0,Z,M)$ is uniformly distributed in X, because exactly four values of X_0 have a given square $X_0^2 \bmod M$.

Theorem P. *Let S be the N-source defined by the muddle-square method on R-bit moduli, and suppose S fails some statistical test A with tolerance $\epsilon \ge 1/2^N$.*

Then we can construct an algorithm F that finds factors of random R-bit Blum integers $M = PQ$ having the form described above, with success probability at least $\epsilon/(4N)$ and with running time $T(F) = O(N^2R^2\epsilon^{-2}T(A) + N^3R^4\epsilon^{-2})$.

Proof. Multiplication mod M can be done in $O(R^2)$ steps; hence $T(f)+T(g) = O(R^2)$. Lemma P4 therefore asserts the existence of a guessing algorithm G with success rate ϵ/N and $T(G) \le T(A)+O(NR^2)$. We can construct G from A using the method of exercise 41. This algorithm G has the property that $s = \mathrm{E}\bigl((-1)^{G(y,z,m)+z\cdot x}\bigr) \ge (\frac12+\epsilon/N)-(\frac12-\epsilon/N) = 2\epsilon/N$, where the expected value is taken over all $(x,z,m) \in X$, and where $(y,z,m) = g(x,z,m)$.

The desired algorithm F proceeds as follows. Given a random $M = PQ$ with unknown P and Q, it computes a random X_0 between 0 and M, and stops immediately with a known factorization if $\gcd(X_0, M) \ne 1$. Otherwise it applies Algorithm L with $G(z) = G(X_0^2 \bmod M, z, M)$ and $k = \lceil \lg(1+2N^2R/\epsilon^2)\rceil$. If one of the 2^k values x output by that algorithm satisfies $x^2 \equiv X_0^2$ (modulo M), there is a 50:50 chance that $x \not\equiv \pm X_0$; then $\gcd(X_0-x, M)$ and $\gcd(X_0+x, M)$ are the prime factors of M. (Cf. Rabin's "SQRT box" in Section 4.5.4.)

The running time of this algorithm is clearly $O(N^2R^2\epsilon^{-2}T(A)+N^3R^4\epsilon^{-2})$, since $\epsilon \ge 2^{-N}$. The probability that it succeeds in factoring M can be estimated as follows. Let $n = |X|/2^R$ be the number of choices of (x,m), and let $s_{xm} = 2^{-R}\sum(-1)^{G(y,z,m)+z\cdot x}$ summed over all R-bit numbers z; thus $s = \sum_{x,m} s_{xm}/n \ge 2\epsilon/N$. Let t be the number of (x,m) such that $s_{xm} \ge \epsilon/N$. The probability that our algorithm deals with such a pair (x,m) is

$$\frac{t}{n} \ge \sum_{x,m}[s_{xm} \ge \epsilon/N]\frac{s_{xm}}{n} = \sum_{x,m}\bigl(1-[s_{xm} < \epsilon/N]\bigr)\frac{s_{xm}}{n}$$
$$\ge \frac{2\epsilon}{N} - \sum_{x,m}[s_{xm} < \epsilon/N]\frac{s_{xm}}{n} \ge \frac{\epsilon}{N}.$$

And in such a case it finds x with probability $\ge \frac12$, by Theorem G, since $2^k > \lceil 2R/s_{xm}^2\rceil$; so it finds a factor with probability $\ge \frac14$. ▮

What does Theorem P imply, from a practical standpoint? Our proof shows that the constant implied by the O is small; let us assume that the running time for factoring is at most $10(N^2R^2\epsilon^{-2}T(A)+N^3R^4\epsilon^{-2})$. Many of the world's greatest mathematicians have worked on the problem of factoring large numbers, especially after factoring was shown to be highly relevant to cryptography in the late 1970s. Since they haven't found a good solution, we have excellent reason to believe that factoring is hard; hence Theorem P will show that $T(A)$ must be large on all algorithms that detect nonrandomness of muddle-square bits.

Long computations are conveniently measured in MIP-years, the number of instructions executed per year by a machine that performs a million instructions per second — namely 31,556,952,000,000 $\approx 3.16 \times 10^{13}$. In 1995, the time to factor a number of 120 decimal digits (400 bits), using the most highly tuned algorithms, was more than 250 MIP-years. The most optimistic researchers who have worked on factorization would be astonished if an algorithm were discovered that requires only $\exp\bigl(R^{1/4}(\ln R)^{3/4}\bigr)$ instructions as $R \to \infty$. But let us assume that such a breakthrough has been achieved, for at least a not-too-small fraction of the R-bit Blum integers M. Then we could factor many numbers of about 50000 bits (15000 digits) in 2×10^{25} MIP-years. If we generate $N = 1000$ random bits by muddle-square with $R = 50000$, and if we assume that all algorithms that are good enough to factor at least $\frac{1}{400000}$ of the 50000-bit Blum integers must run at least 2×10^{25} MIP-years, Theorem P tells us that every such set of 1000 bits will pass all statistical tests for randomness whose running time $T(A)$ is less than 70000 MIP-years: No such algorithm A will be able to distinguish such bits from a truly random sequence with probability $\ge \epsilon = \frac{1}{100}$.

Impressive? No. Such a result is hardly surprising, since we need to specify about 150000 truly random bits just to start up the muddle-square method with X_0, Z, and M when $R = 50000$. Of *course* we should be able to get 1000 random bits back from such an investment!

But in general, the formula becomes

$$T(A) \ge \frac{1}{100000} N^{-2} R^{-2} \exp\bigl(R^{1/4}(\ln R)^{3/4}\bigr) - NR^2,$$

under our conservative assumptions, when $\epsilon = \frac{1}{100}$; the NR^2 term is negligible when R is large. So let's set $R = 200000$ and $N = 10^{10}$. Then we get ten billion pseudorandom muddle-bits from $3R \approx 600000$ truly random bits, passing all statistical tests that require fewer than 7.486×10^{10} MIP-years = 74.86 gigaMIP-years. With $N = 10^{13}$ and $R = 333333$ the number increases to 53.5 teraMIP-years.

The simple pseudorandom generator 3.2.2–(16), which avoids the random mask Z, can also be shown to pass all polynomial-time tests for randomness if factoring is intractable. However, all known asymptotic performance guarantees for the simpler method are somewhat weaker than for muddle-square, currently $O\bigl(RN^4\epsilon^{-4}\log(RN\epsilon^{-1})\bigr)$ versus the $O(N^2R^2\epsilon^{-2})$ of Theorem P. (See exercise 4.5.4–43.)

Everyone believes that there is no factoring algorithm for R-bit numbers whose running time is polynomial in R. If that conjecture is true in a stronger form, so that we cannot even factor $1/R^k$ of the R-bit Blum integers in polynomial time for any fixed k, Theorem P proves that the muddle-square method generates pseudorandom numbers that pass all polynomial-time statistical tests for randomness.

Stating this another way: If you generate random bits with the muddle-square method for suitably chosen N and R, you either get numbers that pass all reasonable statistical tests, or you get fame and fortune for discovering a new factorization algorithm.

G. Summary, history, and bibliography.

▸**Page 165** line 26 ________ 05 Jan 1987

47 $\leadsto$ **46**

Page 166 line 14 ———— 29 Aug 1996

664.] ⇝ 664.] The ideas can also be traced to R. J. Solomonoff, *Information and Control* **7** (1964), 1–22, 224–254; *IEEE Trans.* **IT-24** (1978), 422–432.

Page 166 new paragraphs before the exercises ———— 08 Aug 1995

the topic. ⇝ the topic. Important developments during the next two decades are surveyed in *An Introduction to Kolmogorov Complexity and Its Applications* (Springer, 1993), by Ming Li and Paul M. B. Vitányi.

The foundations of the theory of pseudorandom sequences and effective information were laid by Manuel Blum, Silvio Micali, and Andrew Yao [*FOCS* **23** (1982), 80–91, 112–117; *SICOMP* **13** (1984), 850–864], who constructed the first explicit sequences that pass all feasible statistical tests. Blum and Micali introduced the notion of a "hard-core bit," a Boolean function f such that $f(x)$ and $g(x)$ are easily computed although $f\bigl(g^{-1}(x)\bigr)$ is not; their paper was the origin of Lemma P4. Leonid Levin developed the theory further [*Combinatorica* **7** (1987), 357–363], then he and Oded Goldreich [*STOC* **21** (1989), 25–32] analyzed algorithms such as the muddle-square method and showed that similar use of a mask yields hard-core bits in many further cases. Finally Levin [*J. Symbolic Logic* **58** (1993), 1102–1103] refined the methods of that paper by introducing and analyzing Algorithm L.

Many other authors have contributed to the theory — notably Impagliazzo, Levin, Luby, and Håstad, who showed [*STOC* **21** (1989), 12–24; **22** (1990), 395–404] that pseudorandom sequences can be constructed from any one-way function — but such results are not surveyed here because they apply primarily to abstract complexity theory rather than to practical random number generation. The practical implications of theoretical work on pseudorandomness were first investigated empirically by P. L'Ecuyer and R. Proulx, *Proc. Winter Simulation Conf.* **22** (1989), 467–476.

Page 166 insert new quotation before the exercises ———— 18 Aug 1995

If the numbers are not random,
they are at least higgledy-piggledy.
— GEORGE MARSAGLIA (1984)

Page 167 replacements for exercises 20 and 21 ———— 22 Aug 1995

▸**20.** [*HM30*] (N. G. de Bruijn and P. Erdős.) The first n points of any $[0,1)$ sequence $\langle U_n\rangle$ with $U_0 = 0$ divide the interval $[0,1)$ into n subintervals; let those subintervals have lengths $l_n^{(1)} \ge l_n^{(2)} \ge \cdots \ge l_n^{(n)}$. Clearly $l_n^{(1)} \ge \frac{1}{n} \ge l_n^{(n)}$, because $l_n^{(1)}+\cdots+l_n^{(n)} = 1$. One way to measure the equitability of the distribution of $\langle U_n\rangle$ is to consider

$$\overline{L} = \limsup_{n\to\infty} n l_n^{(1)} \qquad \text{and} \qquad \underline{L} = \liminf_{n\to\infty} n l_n^{(n)}.$$

(a) What are $\overline{L}$ and $\underline{L}$ for van der Corput's sequence (29)?

(b) Show that $l_{n+k-1}^{(1)} \ge l_n^{(k)}$ for $1 \le k \le n$. Use this result to prove that $\overline{L} \ge 1/\ln 2$.

(c) Prove that $\underline{L} \le 1/\ln 4$. [*Hint:* For each n there are numbers $a_1, \ldots, a_{2n}$ such that $l_{2n}^{(k)} \ge l_{n+a_k}^{(n+a_k)}$ for $1 \le k \le 2n$. Moreover, each integer $2, \ldots, n$ occurs at most twice in $\{a_1, \ldots, a_{2n}\}$.]

(d) Show that the sequence $\langle W_n \rangle$ defined by $W_n = \lg(2n+1) \bmod 1$ satisfies $1/\ln 2 > nl_n^{(1)} \ge nl_n^{(n)} > 1/\ln 4$ for all n; hence it achieves the optimum $\overline{L}$ and $\underline{L}$.

21. [*HM40*] (L. H. Ramshaw.) (a) Continuing the previous exercise, is the sequence $\langle W_n \rangle$ equidistributed?

(b) Show that $\langle W_n \rangle$ is the only $[0,1)$ sequence for which we have $\sum_{j=1}^{k} l_n^{(j)} \le \lg(1+k/n)$ whenever $1 \le k \le n$.

(c) Let $\langle f_n(l_1, \ldots, l_n) \rangle$ be any sequence of continuous functions on the sets of n-tuples $\{(l_1, \ldots, l_n) \mid l_1 \ge \cdots \ge l_n \text{ and } l_1 + \cdots + l_n = 1\}$, satisfying the following two properties:

$$f_{mn}(\tfrac{1}{m}l_1, \ldots, \tfrac{1}{m}l_1, \;\ldots\;, \tfrac{1}{m}l_n, \ldots, \tfrac{1}{m}l_n) = f_n(l_1, \ldots, l_n);$$

$$\text{if} \quad \textstyle\sum_{j=1}^{k} l_j \ge \sum_{j=1}^{k} l_j' \text{ for } 1 \le k \le n \quad \text{then} \quad f_n(l_1, \ldots, l_n) \ge f_n(l_1', \ldots, l_n').$$

[Examples are: $nl_n^{(1)}$; $-nl_n^{(n)}$; $l_n^{(1)}/l_n^{(n)}$; $n(l_n^{(1)2} + \cdots + l_n^{(n)2})$.] Let

$$\overline{F} = \limsup_{n\to\infty} f_n(l_n^{(1)}, \ldots, l_n^{(n)})$$

for the sequence $\langle W_n \rangle$. Show that $f_n(l_n^{(1)}, \ldots, l_n^{(n)}) \le \overline{F}$ for all n, with respect to $\langle W_n \rangle$; also $\limsup_{n\to\infty} f_n(l_n^{(1)}, \cdots, l_n^{(n)}) \ge \overline{F}$ with respect to every other $[0,1)$ sequence.

Page 168 replacements for exercises 23 and 24 ________ 11 Dec 1993

23. [*M32*] (a) Show that a $[0,1)$ sequence $\langle U_n \rangle$ is k-distributed if and only if all of the sequences $\bigl\langle (c_1U_n + c_2U_{n+1} + \cdots + c_kU_{n+k-1}) \bmod 1 \bigr\rangle$ are 1-distributed, whenever c_1, c_2, …, c_k are integers not all zero. (b) Show that a b-ary sequence $\langle X_n \rangle$ is k-distributed if and only if all of the sequences $\bigl\langle (c_1X_n + c_2X_{n+1} + \cdots + c_kX_{n+k-1}) \bmod b \bigr\rangle$ are 1-distributed, whenever c_1, c_2, …, c_k are integers with $\gcd(c_1, \ldots, c_k) = 1$.

▸**24.** [*M35*] (J. G. van der Corput.) (a) Prove that the $[0,1)$ sequence $\langle U_n \rangle$ is equidistributed whenever the sequences $\bigl\langle (U_{n+k} - U_n) \bmod 1 \bigr\rangle$ are equidistributed for all $k > 0$. (b) Consequently $\bigl\langle (\alpha_d n^d + \cdots + \alpha_1 n + \alpha_0) \bmod 1 \bigr\rangle$ is equidistributed, when $d > 0$ and α_d is irrational.

Page 169 new exercises ________ 08 Aug 1995

40. [*M26*] Complete the proof of Lemma P1.

41. [*M21*] Lemma P2 shows the existence of a prediction test, but its proof relies on the existence of a suitable k without explaining how we could find k constructively from A. Show that any algorithm A can be converted into an algorithm A' with $T(A') \le T(A) + O(N)$ that predicts B_N from $B_1 \ldots B_{N-1}$ with probability at least $\frac{1}{2} + \bigl(P(A,S) - P(A,\$_N)\bigr)/N$ on any shift-symmetric N-source S.

42. [*M28*] (a) Let X_1, …, X_n be random variables having mean value $\mu = \mathrm{E}\, X_j$ and variance $\sigma^2 = \mathrm{E}\, X_j^2 - (\mathrm{E}\, X_j)^2$ for $1 \le j \le n$. Prove Chebyshev's inequality $\Pr\bigl((X_1 + \cdots + X_n - n\mu)^2 \ge tn\sigma^2\bigr) \le 1/t$, under the additional assumption that $\mathrm{E}(X_iX_j) = (\mathrm{E}\, X_i)(\mathrm{E}\, X_j)$ whenever $i \ne j$.

(b) Let B be a random $k \times R$ binary matrix. Prove that if c and c' are fixed nonzero k-bit vectors, the vectors cB and $c'B$ are independent random R-bit vectors (modulo 2).

(c) Apply (a) and (b) to the analysis of Algorithm L.

43. [*15*] It seems just as difficult to find the factors of any *fixed* R-bit integer M as to find the factors of a *random* R-bit integer. Why then is Theorem P stated for random M instead of fixed M?

[The exercise previously numbered 40 now becomes number 44.]

Page 171 line 4 ——— 26 Aug 1995

with m ... $c = a$. $\leadsto$ with m when m is the computer's word size. Thus we may choose $c = 1$ or $c = a$. Many people have used $c = 0$ together with $m = 2^e$, but they are sacrificing two bits of accuracy and half of the seed values just to save a few nanoseconds of running time (see exercise 3.2.1.2–9).

▸**Page 171** line 11 ——— 13 May 1997

Section 3.4.2 $\leadsto$ Section 3.4.1

Page 171 line 17, extending to page 173 line 11 ——— 26 Aug 1995

The above comments ... were unaware $\leadsto$

viii) At most about $m/1000$ numbers should be generated; otherwise the future will behave more and more like the past. If $m = 2^{32}$, this means that a new scheme (e.g., a new multiplier a) should be adopted after every few million random numbers are consumed.

The comments above apply primarily to machine-language coding. Some of the ideas work fine also in higher-level languages for programming; for example, (1) becomes just '`X=a*X+c`' in the C language, if `X` is of type `unsigned long` and m is the modulus of `unsigned long` arithmetic (usually 2^{32} or 2^{64}). But C gives us no good way to regard `X` as a fraction, as required in (vi) above, unless we convert to double-precision floating-point numbers.

Another variant of (1) is therefore often used in languages like C: We choose m to be a prime number near the largest easily computed integer, and we let a be a primitive root of m; the appropriate increment c for this case is zero. Then (1) can be implemented entirely with simple arithmetic on numbers that remain between $-m$ and $+m$, using the technique of exercise 3.2.1.1–9. For example, when $a = 48271$ and $m = 2^{31} - 1$ (see line 20 of Table 3.3.4–1), we can compute $X \leftarrow aX \bmod m$ with the C code

```
#define MM 2147483647               /* a Mersenne prime */
#define AA 48271  /* this does well in the spectral test */
#define QQ 44488  /* (long)(MM/AA) */
#define RR 3399   /* MM % AA; it is important that RR<QQ */

X=AA*(X%QQ)-RR*(long)(X/QQ);
if (X<0) X+=MM;
```

here `X` is type `long`, and it should be initialized to a nonzero value less than `MM`. The least-significant bits of `X` are now just as random as the most-significant bits, so the precautions of (vi) no longer need to be made.

If you need millions and millions of random numbers, you can combine that routine with another, as in Eq. 3.3.4–(38), by writing some additional code:

```
#define MMM 2147483399           /* a non-Mersenne prime */
#define AAA 40692      /* another spectral success story */
#define QQQ 52774      /* (long)(MMM/AAA) */
#define RRR 3791       /* MMM % AAA; again less than QQQ */
Y=AAA*(Y%QQQ)-RRR*(long)(Y/QQQ);
if (Y<0) Y+=MMM;
Z=X-Y; if (Z<=0) Z+=MM;
```

Like X, variable Y needs to be initially nonzero. This code deviates slightly from 3.3.4–(38) so that the output, Z, is never zero; Z always lies strictly between 0 and 2^{31}. The period length of the Z sequence is about 74 quadrillion, and its numbers now have about twice as many bits of accuracy as the X numbers do.

This method is portable and fairly simple, but not very fast. An alternative scheme based on lagged Fibonacci sequences with subtraction (exercise 3.2.2–23) is even more attractive, because it not only allows easy portability between computers, it is considerably faster, and it delivers random numbers of better quality because the t-dimensional accuracy is probably good for $t \le 100$. Here is a C subroutine *ran_array*(**long** *aa*[ ], **int** *n*) that generates n new random numbers and places them into a given array *aa*, using the recurrence

$$X_j = (X_{j-100} - X_{j-37}) \bmod 2^{30}. \tag{2}$$

This recurrence is particularly well suited to modern computers. The value of n must be at least 100; larger values like 1000 are recommended.

```
#define KK 100                     /* the long lag */
#define LL  37                     /* the short lag */
#define MM (1L<<30)                /* the modulus */
#define mod_diff(x,y) (((x)-(y))&(MM-1))   /* (x-y) mod MM */
long ran_x[KK];                     /* the generator state */
void ran_array(long aa[],int n) {
  register int i,j;
  for (j=0;j<KK;j++) aa[j]=ran_x[j];
  for (;j<n;j++) aa[j]=mod_diff(aa[j-KK],aa[j-LL]);
  for (i=0;i<LL;i++,j++) ran_x[i]=mod_diff(aa[j-KK],aa[j-LL]);
  for (;i<KK;i++,j++) ran_x[i]=mod_diff(aa[j-KK],ran_x[i-LL]);
}
```

All information about numbers that will be generated by future calls to *ran_array* appears in *ran_x*, so you can make a copy of that array in the midst of a computation if you want to restart at the same point later without going all the way back to the beginning of the sequence. The tricky thing about using a recurrence like (2) is, of course, to get everything started properly in the first place, by setting up suitable values of X_0, …, X_{99}. The following subroutine *ran_start*(**long** *seed*) initializes the generator nicely when given any seed number between 0 and $2^{30} - 3 = 1{,}073{,}741{,}821$ inclusive:

```
#define TT  70     /* guaranteed separation between streams */
#define is_odd(x)  ((x)&1)             /* the units bit of x */
#define evenize(x) ((x)&(MM-2))             /* make x even */
void ran_start(long seed) { /* use this to set up ran_array */
  register int t,j;
  long x[KK+KK-1];               /* the preparation buffer */
  register long ss=evenize(seed+2);
  for (j=0;j<KK;j++) {
    x[j]=ss;                        /* bootstrap the buffer */
    ss<<=1; if (ss>=MM) ss-=MM-2;   /* cyclic shift 29 bits */
  }
  for (;j<KK+KK-1;j++) x[j]=0;
  x[1]++;                  /* make x[1] (and only x[1]) odd */
  ss=seed;
  t=TT-1; while (t) {
    for (j=KK-1;j>0;j--) x[j+j]=x[j];           /* "square" */
    for (j=KK+KK-2;j>KK-LL;j-=2) x[KK+KK-1-j]=evenize(x[j]);
    for (j=KK+KK-2;j>=KK;j--) if(is_odd(x[j])) {
      x[j-(KK-LL)]=mod_diff(x[j-(KK-LL)],x[j]);
      x[j-KK]=mod_diff(x[j-KK],x[j]);
    }
    if (is_odd(ss)) {                    /* "multiply by z" */
      for (j=KK;j>0;j--)  x[j]=x[j-1];
      x[0]=x[KK];            /* shift the buffer cyclically */
      if (is_odd(x[KK])) x[LL]=mod_diff(x[LL],x[KK]);
    }
    if (ss) ss>>=1; else t--;
  }
  for (j=0;j<LL;j++) ran_x[j+KK-LL]=x[j];
  for (;j<KK;j++) ran_x[j-LL]=x[j];
}
```

The somewhat curious maneuverings of *ran_start* are explained in exercise 8, which proves that the sequences of numbers generated from different starting seeds are independent of each other: *Every subsequence of 100 consecutive values X_n, X_{n+1}, …, X_{n+99} will be distinct from the subsequences that occur with another seed.* (Strictly speaking, this is known to be true only when $n < 2^{70}$; but there are fewer than 2^{55} nanoseconds in a year.) Several processes can therefore start in parallel with different seeds and be sure that they are doing independent calculations; different groups of scientists working on a problem in different computer centers can be sure that they are not duplicating the work of others if they restrict themselves to different sets of seeds. More than one billion essentially disjoint batches of random numbers are therefore provided by the single routines *ran_array* and *ran_start*. And if that is not enough, you can replace the program parameters 100 and 37 by other values from Table 3.2.2–1.

These C routines use the logical-and operation '`&`' for efficiency, so they are not strictly portable unless the computer uses two's complement representation for integers. Almost all modern computers are based on two's complement arithmetic, but '`&`' is not really necessary for this algorithm. Exercise 9 shows how to get exactly the same sequences of numbers in FORTRAN, using no such tricks. Although the programs illustrated here are designed to generate 30-bit integers, they are easily modified to generate random 52-bit fractions between 0 and 1, on computers that have reliable floating-point arithmetic; see exercise 10.

Caution: The numbers generated by *ran_array* fail the "birthday spacings test" of Section 3.3.2J, and they have other deficiencies that sometimes show up in high-resolution simulations (see exercises 3.3.2–31 and 3.3.2–32). One way to avoid the birthday spacings problem is simply to use only half of the numbers (skipping the odd-numbered elements); but that doesn't cure the other problems. An even better procedure is to follow Martin Lüscher's suggestion, discussed in Section 3.2.2: Use *ran_array* to generate, say, 1009 numbers, but use only the first 100 of these. This method has theoretical support and no known defects. Most users will not need such a precaution, but it is available for applications that demand a high degree of randomness.

A great deal is known about linear congruential sequences like (1), but comparatively little has yet been proved about the randomness properties of lagged Fibonacci sequences like (2). Both approaches seem to be reliable in practice, if they are used with the caveats already stated.

When this chapter was first written in the late 1960s, a truly horrible random number generator called `RANDU` was commonly used on most of the world's computers (see Section 3.3.4). The authors of many contributions to the science of random number generation have often been unaware

Page 173 line 12 ——— 12 Aug 1995

inadequate. $\leadsto$ inadequate. A particularly noteworthy example was the experience of Alan M. Ferrenberg and his colleagues, reported in *Physical Review Letters* **69** (1992), 3382–3384: They tested their algorithms for a three-dimensional problem by considering first a related two-dimensional problem with a known answer, and discovered that supposedly super-quality modern random number generators gave wrong results in the fifth decimal place. By contrast, an old-fashioned run-of-the-mill linear congruential generator, $X \leftarrow 16807X \bmod (2^{31}-1)$, worked fine.

Page 173 lower middle ——— 04 Dec 1986

line 24: 1972–1976 $\leadsto$ 1972–1984

line 25: 102. $\leadsto$ 102; *J. Royal Stat. Soc.* **A149** (1986), 83–107. Subsequent developments are discussed by Shu Tezuka, *Uniform Random Numbers* (Boston: Kluwer, 1995).

Page 173 line 3 before the exercises ______ 07 Aug 1995

quasi-random $\rightsquigarrow$ quasirandom

Page 173 line 2 before the exercises ______ 26 Aug 1995

discussed). $\rightsquigarrow$ discussed). The origins of Monte Carlo methods for computers are discussed by N. Metropolis and R. Eckhardt in *Stanislaw Ulam 1909–1984*, a special issue of *Los Alamos Science* **15** (1987), 125–137.

Page 176 new paragraph before line −2 of exercise 5 ______ 02 Aug 1995

In the summer of 1952, Christopher Strachey had used the hardware random number generator of the Ferranti Mark I to compose the following letter:

> Honey Dear
>
> My sympathetic affection beautifully attracts your affectionate enthusiasm. You are my loving adoration: my breathless adoration. My fellow feeling breathlessly hopes for your dear eagerness. My lovesick adoration cherishes your avid ardour.
>
> Yours wistfully,
> M. U. C.

[*Encounter* **3** (1954), 4, 25–31; another example appeared in the article on Electronic Computers in the 64th edition of *Pears Cyclopedia* (London, 1955), 190–191.]

Page 177 new exercise ______ 26 Aug 1995

8. [*HM32*] Let X_0, X_1, ... be the numbers produced by *ran_array* after *ran_start* has initialized the generation process with seed s, and consider the polynomials

$$P_n(z) = X_{n+62}z^{99} + X_{n+61}z^{98} + \cdots + X_n z^{37} + X_{n+99}z^{36} + \cdots + X_{n+64}z + X_{n+63}.$$

(a) Prove that $P_n(z) \equiv z^{h(s)-n}$ (modulo 2 and $z^{100}+z^{37}+1$), for some exponent $h(s)$. (b) Express $h(s)$ in terms of the binary representation of s. (c) Prove that if X'_0, X'_1, ... is the sequence of numbers produced by the same routines from the seed $s' \neq s$, we have $X_{n+k} \equiv X'_{n'+k}$ (modulo 2) for $0 \le k < 100$ only if $|n-n'| \ge 2^{70}-1$.

Page 177 new exercises ______ 27 Aug 1995

9. [*22*] Convert the C code for *ran_array* and *ran_start* to FORTRAN 77 subroutines that generate exactly the same sequences of numbers.

▸**10.** [*M25*] Assuming that floating-point arithmetic on numbers of type `double` is properly rounded in the sense of Section 4.2.2 (hence exact when the values are suitably restricted), convert the C routines *ran_array* and *ran_start* to similar programs that deliver double-precision random fractions in the range $[0,1)$, instead of 30-bit integers.

Page 177 another new exercise ______ 30 Aug 1995

11. [*M21*] What random number generator would be suitable for a minicomputer that does arithmetic only on integers in the range $[-32768\,..\,32767]$?

Page 177 yet another ______ 19 Sep 1995

▸**12.** [*M25*] Compare the subtract-with-borrow generators of exercise 3.2.1.1–11 to the lagged Fibonacci generators implemented in the programs of this section.

Page 178 line 15 ______ 16 Apr 1995

MRS. LA TOUCHE (19th century) $\rightsquigarrow$ M. P. LA TOUCHE (1878)

Page 181 bottom line ______ 01 Jan 1997

Tsu Chhung-Chih (who died c. 500 A.D.) $\rightsquigarrow$ Tsu Ch'ung-Chih (who died in A.D. 501)

Page 182 line 2 ______ 01 Jan 1997

1 chhih, 4 tshun, 1 fên $\rightsquigarrow$ 1 ch'in, 4 ts'un, 1 fen

▸**Page 183** line 4 ______ 21 Feb 1992

ex institute hominum $\rightsquigarrow$ ex instituto hominum

Page 183 bottom ______ 16 Apr 1995

line -7 to -5: 454. The first . . . Caramuel $\rightsquigarrow$ 454; Harriot's discovery of binary notation was first cited by Frank Morley in *The Scientific Monthly* **14** (1922), 60–66.

The first published treatment of the binary system appeared in the work of a prominent Cistercian bishop, Juan de Caramuel

lines -2 and -1: (except for the trivial operation of adding unity) $\rightsquigarrow$ except in the sexagesimal case

Page 184 line -6 ______ 16 Apr 1995

rector of a parish in Maryland $\rightsquigarrow$ professor at the College of William and Mary

Page 185 line 5 ______ 16 Apr 1995

`B0160` $\rightsquigarrow$ `C0160`

Page 185 line 23 ______ 27 Sep 1995

(privately printed, 1971) $\rightsquigarrow$ (Los Angeles: Tomash, 1981)

Page 187 line 10 from the bottom ______ 27 Sep 1995

notations. $\rightsquigarrow$ notations. Two's complement arithmetic on n-bit numbers is arithmetic modulo 2^n; ones' complement arithmetic is modulo $2^n - 1$.

Page 190 caption ______ 30 Sep 1996

The set S. $\rightsquigarrow$ The twin-dragon.

Page 190 line -9 ______ 30 Sep 1996

Figure 1 $\rightsquigarrow$ It is also known as the "twin-dragon fractal" [see M. F. Barnsley, *Fractals Everywhere*, second edition (Academic Press, 1993), 306, 310]. Figure 1

Page 192 near the top ______ 13 Dec 1996

line 4: book. $\rightsquigarrow$ book, and by Ṭabarī in Persia more than 100 years before Fibonacci.
line 5: 3.4. $\rightsquigarrow$ 3.4; H. Hermelink, *Janus* **65** (1978), 105–117.

Page 193 line 20 ______ 26 Sep 1996

binomial $\rightsquigarrow$ combinatorial

Page 195 last line of exercise 23 ——— 02 Apr 1981

zero. $\leadsto$ zero, if $0 \in D$. Show that this conclusion need not be true if $0 \notin D$.

Page 197 line 2 ——— 09 Oct 1995

invented about 1900 by K. Hensel. $\leadsto$ introduced by K. Hensel in *Crelle* **127** (1904), 51–84.

Page 197 bottom line ——— 18 Mar 1995

algorithm $\leadsto$ algorithm by showing that k_n is a Fibonacci number

Page 197 new exercise ——— 24 Aug 1996

▸**34.** [*22*] (G. W. Reitwiesner, 1960.) Explain how to represent a given integer n in the form $(\ldots a_2a_1a_0)_2$, where each a_j is -1, 0, or 1, using the fewest nonzero digits.

Page 198 Planck's constant, not Dirac's ——— 10 Jan 1981

line 21: $\hbar = 1.0545 \leadsto h = 6.6256$
line -3: $\hbar = (24, +.10545000) \leadsto h = (24, +.66256000)$

Page 210 line -4 ——— 12 Jan 1981

computer System $\leadsto$ *Computer System*

Page 211 line 2 . . . . 27 Sep 1995

I will of course revise this section extensively to reflect ANSI/IEEE Standard 754, which revolutionized floating-point hardware after the second edition of this book was written. Relevant references are: IEEE Micro **4** *(1984), 86–100; W. J. Cody, Comp. Sci. and Statistics: Symp. on the Interface* **15** *(1983), 133–139; W. M. Kahan, Mini/Micro West-83 Conf. Record (1983), paper 16/1; D. Goldberg, Computing Surveys* **23** *(1991), 5–48, 413; W. J. Cody and J. T. Coonen, ACM Trans. Math. Software* **19** *(1993), 443–451.*

▸**Page 214** line 3 after (2) ——— 26 Apr 1985

satified $\leadsto$ satisfied

▸**Page 216** replacement for (18) ——— 12 Jan 1981

$$|\delta(x)| = \frac{|\rho(x)|}{|x|} \le \frac{|\rho(x)|}{b^{e-1} + |\rho(x)|} \le \frac{\frac{1}{2}b^{e-p}}{b^{e-1} + \frac{1}{2}b^{e-p}} < \tfrac{1}{2}b^{1-p}. \qquad (18)$$

Page 218 strengthen (32) ——— 27 Apr 1981

$(\epsilon_1 + \epsilon_2) \leadsto \bigl(\min(\epsilon_1, \epsilon_2)\bigr)$

▸**Page 221** line 8 ——— 22 May 1997

$t + r$ and $r \leadsto t + r$ and t

Page 222 lines 23–26 ——— 08 Jan 1981

line 23: but if $\leadsto$ if
line 24: occur. [Roy $\leadsto$ occur, although repeated rounding of a number like 2.5454 will lead to almost as much error. [Cf. Roy
line 25: On the other hand, some $\leadsto$ Some

line 26: remainder $\rightsquigarrow$ least significant digit
line 26: often. $\rightsquigarrow$ often. Exercise 23 demonstrates this advantage of round-to-even.

▸**Page 223** line -14 ——— 08 Jul 1992

precison $\rightsquigarrow$ precision

Page 223 Planck's constant, not Dirac's ——— 10 Jan 1981

line -17: $(-23, +.00010545) \rightsquigarrow (-23, +.00066256)$
line -16: $(-26, +.66256000)$
line -10: $(0, +.0063507) \rightsquigarrow (1, +.00039903)$
page 225 line -1 becomes: $h = \bigl[(-26, +.66252000), (-26, +.66261000)\bigr]$
page 226 line 2: $\hbar \rightsquigarrow h$
page 226 line 3 becomes: $N \otimes h = \bigl[(-2, +.39898544), (-2, +.39907676)\bigr]$

Page 227 new paragraph to follow line 23 ——— 22 Aug 1996

An extension of the Pascal language that allows variables to be of type "interval" was developed at the University of Karlsruhe in the early 1980s. For a description of this language, which also includes numerous other features for scientific computing, see *Pascal-SC* by Bohlender, Ullrich, Wolff von Gudenberg, and Rall (Academic Press, 1987).

Page 227 corrections to bad German ——— 22 Apr 1981

line 24: *Begrund-* $\rightsquigarrow$ *Begrün-*
line 25: *ung der Rechenarithmetik* $\rightsquigarrow$ *dung der Rechnerarithmetik*

Page 227 last line before exercises ——— 16 Jun 1981

1980 $\rightsquigarrow$ 1981

Page 230 new exercise ——— 22 Aug 1996

28. [*M25*] (U. Kulisch.) When the polynomial $2y^2 + 9x^4 - y^4$ is evaluated for $x = 408855776$ and $y = 708158977$ using standard 53-bit double-precision floating point arithmetic, the result is $\approx -3.7 \times 10^{19}$. Evaluating it in the alternate form $2y^2 + (3x^2 - y^2)(3x^2 + y^2)$ gives $\approx +1.0 \times 10^{18}$. The true answer, however, is 1.0 (exactly). Explain how to construct similar examples of numerical instability.

Page 230 new exercise ——— 17 Apr 1995

29. [*M21*] Given that u, u', v, and v' are positive floating point numbers with $u \sim u'$ (ϵ) and $v \sim v'$ (ϵ), prove that there's a small ϵ' such that $u \oplus v \sim u' \oplus v'$ (ϵ'), assuming normalized arithmetic.

Page 231 and following . . . . 29 Oct 1995

The material of Section 4.2.3 is no longer of interest, so I will eliminate it from the next edition. The few remaining nuggets of relevant information, like the comments about floating-point cosine or the topic of exercise 9, can be moved to Section 4.2.2.

▸**Page 246** broken type in (23) ——— 14 Jan 1985

$c_{m--1} \rightsquigarrow c_{m-1}$

Page 247 lines −9 through −6 ——— 27 Sep 1995

survey by Ralph A. Raimi ... 144. ⇝ surveys by Ralph A. Raimi, *AMM* **83** (1976), 521–538, and Peter Schatte, *J. Information Processing and Cybernetics* **24** (1988), 443–455, for a comprehensive review of the literature.

Page 247 new paragraph for bottom of page ——— 27 Sep 1995

Floating-point computations operate primarily on non-integer numbers, of course; we have studied integers because of their familiarity and their simplicity. When arbitrary real numbers are considered, theoretical results are more difficult to obtain, but evidence is accumulating that the same statistics apply, in the sense that repeated calculations with real numbers will nearly always tend to yield better and better approximations to a logarithmic distribution of fraction parts. For example, Peter Schatte [*Zeitschrift für Angewandte Math. und Mechanik* **53** (1973), 553–565] showed that, under mild restrictions, the products of independent, identically distributed random real variables approach the logarithmic distribution. And the sums of such variables do too, but only in the sense of repeated averaging. Similar results have been obtained by J. L. Barlow and E. H. Bareiss, *Computing* **34** (1985), 325–347.

Page 249 line 1 of exercise 17 ——— 28 Sep 1995

(R. L. Duncan.) ⇝ (M. Tsuji.)

Page 249 new exercise ——— 27 Sep 1995

19. [*HM25*] (R. L. Duncan.) Prove that the leading digits of Fibonacci numbers obey the logarithmic law of fraction parts: $\Pr(10f_{F_n} < r) = \log_{10} r$.

Page 250 line 15 ——— 28 Jun 1995

any integer ⇝ any nonnegative integer

Page 250 near line 30 . . . . 28 Jun 1995

In the next edition I'll delete the sentence "For convenience ... section." On page 251, lines 23–24, "the reader ... sort" will become "we will discuss the use of negative numbers and complement notations near the end of this section". The discussion of complement notations on page 261 will then be modified appropriately. (The new machine, `MMIX`*, uses two's complement notation, like almost all other computers currently made.)*

Page 251 change in notation . . . . 28 Jun 1981

Throughout this section I should write, for example, $(u_{n-1}\ldots u_1u_0)_b$ *instead of* $(u_1u_2\ldots u_n)_b$.

▸**Page 251** bottom line ——— 01 Sep 1987

is a valid ⇝ is valid

Page 258 first three lines of step D3 ——— 28 Sep 1995

If $u_j = v_1$, ... latter test determines ⇝ Set $\hat q \leftarrow \lfloor (u_jb + u_{j+1})/v_1 \rfloor$ and $\hat r \leftarrow (u_jb+u_{j+1}) \bmod v_1$. Now test if $\hat q = b$ or $v_2\hat q > b\hat r + u_{j+2}$; if so, decrease $\hat q$ by 1, increase $\hat r$ by v_1, and repeat this test if $\hat r < b$. [The test on v_2 determines

Page 260 line −6 ——— 05 May 1996

Further $\leadsto$ When the radix b is comparatively small, so that b^2 is less than the computer's word size, multiprecision division can be speeded up by not reducing individual digits of intermediate results to the range $[0\,..\,b)$; see D. M. Smith, *Math. Comp.* **65** (1996), 157–163. Further

▸**Page 263** replacement for line 3 ——— 26 May 1997

$$\frac{\partial^2 G}{\partial z^2} = \sum_n G_n''(z)t^n = \frac{-t^2(b+1-2t)}{p(z,t)^2} + \frac{t^2(b-zt)(b+1-2t)^2}{2p(z,t)^3}.$$

▸**Page 264** line −9 ——— 26 May 1997

4.3.3D $\leadsto$ 4.3.3R

▸**Page 264** line −4 ——— 25 Mar 1982

Eq. 4.2.3–3 $\leadsto$ Eq. 4.2.3–(2)

Page 265 line 8 ——— 05 May 1996

57–81. $\leadsto$ 57–81; D. M. Smith, *ACM Trans. Math. Software* **17** (1991), 273–283.

Page 265 line 21 ——— 28 Jun 1995

3–5. $\leadsto$ 3–5. See also Section 4.3.3E.

▸**Page 268** exercise 36 line 1 ——— 15 Jan 1981

Appendix B $\leadsto$ Appendix A

Page 268 exercise 36 line 5 ——— 15 Mar 1995

in 1974.] $\leadsto$ in 1973. The billion-digit barrier was passed in 1989 by Gregory V. Chudnovsky and David V. Chudnovsky, and independently by Yasumasa Kanada and Yoshiaki Tamura. The Chudnovskys extended their calculation to two billion digits in 1991, after 250 hours of computation on a home-built parallel machine; see Richard Preston, *The New Yorker* **68**, 2 (2 March 1992), 36–67. The novel formula they used is described in *Proc. Nat. Acad. Sci.* **86** (1989), 8178–8182.]

Page 268 new exercise ——— 07 Jan 1984

38. [*M35*] Suppose u and v are integers in the range $0 \le u, v < 2^n$. Devise a way to compute the geometric mean $\lfloor\sqrt{uv} + \frac{1}{2}\rfloor$ by doing $O(n)$ operations of addition, subtraction, and comparison of $(n+2)$-bit numbers. [*Hint:* Use a "pipeline" to combine the classical methods of multiplication and square rooting.]

Page 268 new exercise ——— 19 Apr 1995

39. [*M25*] Explain how to compute $\ln\phi$ to high precision, given a suitably precise approximation to ϕ, using only multiprecision addition, subtraction, and division by small numbers.

Page 268 new exercise ——— 17 Jan 1996

40. [*25*] (D. Bailey, P. Borwein, and S. Plouffe, 1996.) Explain how to compute the nth bit of the binary representation of π without knowing the previous $n-1$ bits, by using the identity

$$\pi = \sum_{k\geq 0} \frac{1}{16^k}\left(\frac{4}{8k+1} - \frac{2}{8k+4} - \frac{1}{8k+5} - \frac{1}{8k+6}\right)$$

and doing $O(n \log n)$ arithmetic operations on $O(\log n)$-bit integers. (Assume that the binary digits of π do not have surprisingly long stretches of consecutive 0s or 1s.)

Page 268 new exercise ——— 25 Aug 1996

41. [*M24*] Sometimes we want to divide u by v when we know that the remainder will be zero. Show that if u is a $2n$-place number and v is an n-place number with $u \bmod v = 0$, we can save about 75% of the work of Algorithm D if we compute half of the quotient from left to right and the other half from right to left.

Page 268 new exercise ——— 28 Jun 1995

▸**42.** [*M26*] Many applications of high-precision arithmetic require repeated calculations modulo a fixed n-place number w, where w is relatively prime to the base b. We can speed up such calculations by using a trick due to Peter L. Montgomery [*Math. Comp.* **44** (1985), 519–521], which streamlines the remaindering process by essentially working from right to left instead of left to right.

(a) Given $u = \pm(u_{m+n-1} \ldots u_1 u_0)_b$, $w = (w_{n-1} \ldots w_1 w_0)_b$, and a number w' such that $w_0 w' \bmod b = 1$, show how to compute $v = \pm(v_{n-1} \ldots v_1 v_0)_b$ such that $b^m v \bmod w = u \bmod w$.

(b) Given n-place signed integers u, v, w with $|u|, |v| < w$, and given w' as in (a), show how to calculate an n-place integer t such that $|t| < w$ and $b^n t \equiv uv$ (modulo w).

(c) How do the algorithms of (a) and (b) facilitate arithmetic mod w?

Page 268 new exercise ——— 12 Apr 1996

▸**43.** [*22*] Shades of gray or components of color values in digitized images are usually represented as 8-bit numbers u in the range $[0\,..\,255]$, denoting the fraction $u/255$. Given two such fractions $u/255$ and $v/255$, graphical algorithms often need to compute their approximate product $w/255$, where w is the nearest integer to $uv/255$. Prove that w can be obtained from the efficient formula

$$t = uv + 128, \qquad w = \bigl\lfloor (\lfloor t/256 \rfloor + t)/256 \bigr\rfloor.$$

Page 271 top ——— 04 Oct 1995

lines 3–6: [See ... apparently first $\leadsto$ Mathematicians in medieval India developed the techniques further, with their methods of *kuṭṭaka* (see Section 4.5.2), but Theorem C was first

line 7: (1247). Numerous $\leadsto$ (1247); this work considers also the case where the moduli might have common factors as in exercise 3. [See J. Needham, *Science and Civilization in China* **3** (Cambridge University Press, 1959), 33–34, 119–120; Y. Li and S. Du, *Chinese Mathematics* (Oxford: Clarendon, 1987), 92–94, 105, 161–166; K. Shen, *Archive for History of Exact Sciences* **38** (1988), 285–305.] Numerous

line 9: (New York: Chelsea, 1952) $\leadsto$ (Carnegie Inst. of Washington, 1920)

Page 272 bottom four lines ———— 02 Oct 1995

$m_1 = 2^{35} - 1, \ldots 2^{161}$. ⇝ $m_1 = 2^{32} - 1$, $m_2 = 2^{31} - 1$, $m_3 = 2^{29} - 1$, $m_4 = 2^{27} - 1$, $m_5 = 2^{25} - 1$, on a computer with word size 2^{32}; this would permit efficient addition, subtraction, and multiplication of integers in a range of size $m_1m_2m_3m_4m_5 > 2^{143}$.

Page 273 line −2, through page 274 line 2 ———— 02 Oct 1995

Furthermore, ... Eq. (9) requires ⇝ There are much better ways to compute M_j than to use (8); in this respect we can see again the distinction between mathematical elegance and computational efficiency. But even if we find M_j by the best possible method, we're stuck with the fact that it is a multiple of the huge number m/m_j. Thus, (9) forces us to do

▸**Page 274** line −3 ———— 25 Nov 1981

1965 ⇝ 1964

Page 276 near the top ———— 02 Oct 1995

line 6: 107–112]. ⇝ 107–112], A. S. Fraenkel and D. Loewenthal [*J. Res. Nat. Bureau of Standards* **75B** (1971), 67–75].

line 8: one hour's running time ⇝ 20 minutes

Page 276 line 29 ———— 03 Jul 1981

Rev. Romaine des Math. ⇝ *Rev. Roumaine de Math. Pures et Appl.*

Page 278 new exercise ———— 01 Jan 1997

▸**14.** [*M25*] (*Mersenne multiplication.*) The cyclic convolution of $(x_0, x_1, \ldots, x_{n-1})$ and $(y_0, y_1, \ldots, y_{n-1})$ is defined to be $(z_0, z_1, \ldots, z_{n-1})$, where

$$z_k = \sum_{i+j \equiv k \,(\mathrm{modulo}\, n)} x_i y_j, \qquad \text{for } 0 \le k < n.$$

We will study efficient algorithms for cyclic convolution in Sections 4.3.3 and 4.6.4.

Consider q-bit integers u and v that are represented in the form

$$u = \sum_{k=0}^{n-1} u_k 2^{\lfloor kq/n \rfloor}, \qquad v = \sum_{k=0}^{n-1} v_k 2^{\lfloor kq/n \rfloor},$$

where $0 \le u_k, v_k < 2^{\lfloor (k+1)q/n \rfloor - \lfloor kq/n \rfloor}$. (This representation is a mixture of radices $2^{\lfloor q/n \rfloor}$ and $2^{\lceil q/n \rceil}$.) Suggest a good way to find the representation of

$$w = (uv) \bmod (2^q - 1),$$

using an appropriate cyclic convolution. [*Hint:* Do not be afraid of floating-point arithmetic.]

Page 279 line 17 ———— 16 Sep 1995

is large. ⇝ is large. Exercise 16 discusses an implementation of this approach.

▸**Page 280** line −4 ———— 10 Nov 1984

Soviet Mathematics **3** ⇝ *Soviet Mathematics* **4**

Page 281 and following . . . . 02 Oct 1995

The Toom-Cook algorithm no longer deserves seven pages of discussion, so I'll reduce this to a page or so in the next edition. I'll probably also eliminate the exposition of Schönhage's old modular method, except perhaps to make an exercise for Section 4.3.2 out of (30). *(Note that the present answer to exercise 4.3.2–7 cites Schönhage's technique.)*

▸**Page 282** line −13 ——— 17 Apr 1984

$\theta_{m-1}(x+m+2) \leadsto \theta_{m-1}(x-m+2)$

Page 283 new name . . . . 30 May 1997

In the next edition, Algorithm C will be renamed Algorithm T.

Page 286 new name . . . . 30 May 1997

In the next edition, Theorem C will be renamed Theorem B.

▸**Page 288** in displayed equation (23) ——— 08 Jan 1982

$T(n) = O(N^{\log_3 6}) = O(N^{1.63}) \leadsto T(n) = O(n^{\log_3 6}) = O(n^{1.63})$

Page 290 line −8 ——— 04 Oct 1995

In order $\leadsto$ Similar ideas, but with all-integer methods, had been worked out independently by J. M. Pollard [*Math. Comp.* **25** (1971), 365–374]. In order

Page 291 lines 3, 5, and 8 ——— 31 Oct 1995

Delete the '$\cdot$' signs; e.g.,

$$\omega^{(s_{k-1}0\ldots0)_2} \cdot A^{[0]}(1, t_{k-2}, \ldots, t_0) \qquad \leadsto \qquad \omega^{(s_{k-1}0\ldots0)_2} A^{[0]}(1, t_{k-2}, \ldots, t_0)$$

▸**Page 291** middle ——— 14 Jan 1997

in (33): $\omega^{(s_0 s_1 \ldots s_{k-1})_2 \cdot (t_{k-1} \ldots t_{k-j} 0 \ldots 0)_2} u_t \leadsto \omega^{2^{k-j}(s_{k-j} \ldots s_{k-1})_2 (t_{k-1} \ldots t_{k-j})_2} u_t$
and on the next line: so that $\leadsto$ where $t = (t_{k-1} \ldots t_1 t_0)_2$, so that

Page 292 replacement for bottom two lines ——— 29 Feb 1996

have $\omega_{r+1} = x_{r+1} + i y_{r+1}$ where

$$x_{r+1} = \sqrt{\frac{1+x_r}{2}}, \qquad y_{r+1} = \frac{y_r}{2x_{r+1}}. \tag{39}$$

[Cf. S. R. Tate, *IEEE Transactions* **SP-43** (1995), 1709–1711.]

▸**Page 293** lines 4 and 5 ——— 17 Apr 1984

$2^{K-r} \leadsto 2^{k-r}$ (twice)

Page 293 line 3 before (41) ——— 29 Feb 1996

rounding. $\leadsto$ rounding, because $x_{r+1}^2 + y_{r+1}^2 = (1 + x_r^2 + y_r^2 + 2x_r)/(2 + 2x_r)$ in (39).

Page 294 line −2 ——— 07 Sep 1989

multiplication, even though . . . behavior. $\leadsto$ multiplication.

Page 294 in general . . . . 02 Oct 1995

Before I complete the next edition of Section 4.3.3, I need to gain personal experience implementing this Fourier transform algorithm and the all-integer method of exercise 4.6.4–59. The discussion I give now is just armchair theory, not really tested with practical constraints. For example, it seems that my proposed implementation of the Schönhage-Strassen method requires about $120n$ *bits of memory to multiply two* n*-bit numbers, or* $60n$ *bits with the suggestions of answer 4.6.4–14.*

Page 295 line 22 ______ 13 Jul 1996

lookup. $\rightsquigarrow$ lookup. (The time to access a word of memory is assumed to be proportional to the number of bits in the address of that word.)

Page 295 new paragraph to follow line 28 ______ 21 Aug 1996

An unusual general-purpose computer called Little Fermat, with a special ability to multiply large integers rapidly, was designed in 1986 by D. V. Chudnovsky, G. V. Chudnovsky, M. M. Denneau, and S. G. Younis. Its hardware featured fast arithmetic modulo $2^{256}+1$ on 257-bit words; a convolution of 256-word arrays could than be done using 256 single-word multiplications, together with three discrete transforms that required only addition, subtraction, and shifting. This made it possible to multiply two 10^6-bit integers in less than 0.1 second, based on a pipelined cycle time of approximately 60 nanoseconds [*Proc. Third Int. Conf. on Supercomputing* **2** (1988), 498–499; *Contemporary Math.* **143** (1993), 136].

Page 299 bottom line ______ 07 Aug 1995

to appear $\rightsquigarrow$ *STOC* **13** (1981), 308–311; Mansour, Nisan, and Tiwari, *STOC* **22** (1990), 235–243.

Page 301 new rating for exercise 12 ______ 03 Oct 1995

[*30*] $\rightsquigarrow$ [*40*]

Page 301 line 7 of exercise 15 ______ 26 Mar 1982

hence also on pointer machines. $\rightsquigarrow$ and is $O(n \log n)$ on pointer machines.

Page 301 new exercise ______ 16 Sep 1995

16. [*25*] Devise a scheme to allocate memory for the intermediate results when multiplication is performed by a recursive algorithm based on (2): Given two n-place integers u and v, each in the least significant n places of $4n$-place workspace arrays, show how to arrange the computation so that the product uv appears in the most significant $2n$ places of the u array.

Page 301 new exercise ______ 10 Sep 1995

▸**17.** [*M22*] Show how to compute $uv \bmod m$ with a bounded number of operations that meet the ground rules of exercise 3.2.1.1–10, if you are also allowed to test whether one operand is less than another.

Page 305 new Table 1 and the preceding/following lines ———— 30 Sep 1982

following table. ... $(2d-10)+2^4$ ⇝ following table suggested by Peter L. Montgomery.

Table 1

DOUBLING A BINARY-CODED DECIMAL NUMBER

Operation	*General form*	*Example*
1. Given number	$u_{11}\,u_{10}\,u_9\,u_8\;\,u_7\,u_6\,u_5\,u_4\;\,u_3\,u_2\,u_1\,u_0$	$0011\,0110\,1001 = 3\,6\,9$
2. Add 3 to each digit	$v_{11}\,v_{10}\,v_9\,v_8\;\,v_7\,v_6\,v_5\,v_4\;\,v_3\,v_2\,v_1\,v_0$	$0110\,1001\,1100$
3. Extract each high bit	$v_{11}\;\,0\;\,0\;\,0\;\,v_7\;\,0\;\,0\;\,0\;\,v_3\;\,0\;\,0\;\,0$	$0000\,1000\,1000$
4. Shift right 2 and subtract	$0\;\,v_{11}\,v_{11}\,0\;\,0\;\,v_7\,v_7\;\,0\;\,0\;\,v_3\,v_3\;\,0$	$0000\,0110\,0110$
5. Add original number	$w_{11}w_{10}w_9w_8\,w_7w_6w_5w_4\,w_3w_2w_1w_0$	$0011\,1100\,1111$
6. Add original number	$x_{12}\,x_{11}\,x_{10}\,x_9\,x_8\;\,x_7\,x_6\,x_5\,x_4\;\,x_3\,x_2\,x_1\,x_0$	$0\,0111\,0011\,1000 = 7\,3\,8$

This method changes each individual digit d into $2d$ when $0 \le d \le 4$, and into $6+2d=(2d-10)+2^4$

▸**Page 305** line −5 ———— 18 Jan 1981

$5 \le d \le q$ ⇝ $5 \le d \le 9$

▸**Page 307** last line of Example 1 ———— 27 Jan 1981

$(14198757)_{10}$ ⇝ $(1419857)_{10}$

▸**Page 310** line −3 of exercise 1 ———— 07 Sep 1989

$3b_2b_1b_0+2b_1b_0+12b_0+37$ ⇝ $3b_2b_1b_0+9b_1b_0+12b_0+37$

▸**Page 315** line 4 ———— 30 Sep 1982

$0 \le v < 256$ ⇝ $0 \le u' < 256$

Page 315 line 1 before the exercises ———— 06 Oct 1995

111. ⇝ 111; *IEEE Trans.* **C-32** (1983), 378–388; *IEEE Trans.* **C-34** (1985), 3–18; *IEEE Trans.* **C-39** (1990), 1106–1115.

Page 318 top ———— 25 Jun 1996

line 2: Scholars ⇝ Some scholars
line 5: We might call it ⇝ However, very little hard evidence about such early history has survived [see W. R. Knorr, *The Evolution of the Euclidean Elements* (Dordrecht: 1975)].

We might call Euclid's method

Page 318 line 5 ———— 20 Apr 1995

We might ⇝ Aristotle hinted at it in his *Topics*, 158b, 29–35. We might

Page 321 line 4 ——— 15 Dec 1996

discovered $\leadsto$ devised

▸**Page 323** line 26 ——— 16 Jun 1997

$B = 0 \ldots\ 12n + 8 \leadsto B = 1$, $C = n$, $D = 2n - 2$, $E = n - 1$; this amounts to $13n + 8$

Page 323 new material to precede line −10 ——— 15 Dec 1996

The binary gcd algorithm itself might have a distinguished pedigree, since it may well have been known in ancient China. Chapter 1, Section 6 of a classic text called *Chiu Chang Suan Shu*, the "Nine Chapters on Arithmetic" (c. 1st century A.D.), gives the following method for reducing a fraction to lowest terms:

> If halving is possible, take half.
>
> Otherwise write down the denominator and the numerator, and subtract the smaller from the greater.
>
> Repeat until both numbers are equal.
>
> Simplify with this common value.

If the repeat instruction means to go back to the halving step instead of to repeat the subtraction step — this point isn't clear — the method is essentially Algorithm B. [See Y. Mikami, *The Development of Mathematics in China and Japan* (Leipzig: 1913), 11; K. Vogel, *Neun Bücher Arithmetischer Technik* (Braunschweig: Vieweg, 1968), 8.]

Page 325 line −11 ——— 11 Oct 1995

The validity of Algorithm X $\leadsto$ Algorithm X can be traced to the fifth-century work *Aryabhatiya* by Aryabhata. His description was rather cryptic, but later commentators such as Bhâscara I in the sixth century clarified the rule, which was called *kuṭṭaka* ("the pulverizer"). [See B. Datta and A. N. Singh, *History of Hindu Mathematics* **2** (Lahore, 1938), 89–116.] Its validity

Page 327 line −14 ——— 07 Oct 1995

scope of this book. $\leadsto$ scope of this book. [See Henri Cohen, *A Course in Computational Algebraic Number Theory* (Springer, 1993), Chapter 2.]

Variants of Euclid's algorithm can be used also with Gaussian integers $u+iu'$ and in certain other quadratic number fields. See, for example, E. Kaltofen and H. Rolletschek, *Math. Comp.* **53** (1989), 697–720; A. Knopfmacher and J. Knopfmacher, *BIT* **31** (1991), 286–292.

Page 330 lines 18–19 ——— 07 Oct 1995

advantage that ... furthermore, it determines $\leadsto$ advantage that it determines

Page 332 and page 333 . . . . 06 Oct 1995

This derivation of the solution to (34) *should be condensed to a few lines.*

Page 337 new rating for exercise 14 ——— 11 Oct 1995

[*M26*] $\leadsto$ [*M21*]

▸**Page 338** line 12 ________ 16 Jun 1997

$1 \le n \le m \le N$ ⇝ $1 \le n < m \le N$

Page 338 new rating for exercise 28 ________ 29 Oct 1995

[*M37*] ⇝ [*M32*]

Page 339 new exercise ________ 07 Aug 1995

▸**37.** [*M25*] (R. P. Brent and H. T. Kung.) The following variant of the binary gcd algorithm is better than Algorithm B from the standpoint of hardware implementation, because it does not require testing the sign of $u - v$. Assume that u is odd; u and v can be either positive or negative.

C1. [Initialize.] Set $c \leftarrow 0$. (This counter estimates the difference between $\lg|u|$ and $\lg|v|$.)

C2. [Done?] If $v = 0$, terminate with $|u|$ as the answer.

C3. [Make v odd.] Set $v \leftarrow v/2$ and $c \leftarrow c + 1$ zero or more times, until v is odd.

C4. [Make $c \le 0$.] If $c > 0$, interchange $u \leftrightarrow v$ and set $c \leftarrow -c$.

C5. [Reduce.] Set $w \leftarrow (u+v)/2$. If w is even, set $v \leftarrow w$; otherwise set $v \leftarrow w - v$. Return to step C2. ▮

Prove that step C2 is performed at most $2 + 2\lg\max(|u|, |v|)$ times.

Page 339 lines 5–6 after (1) ________ 08 Jul 1996

J. Tropfke, ... 74–84, ⇝ Claude Brezinski, *History of Continued Fractions and Padé Approximants* (Springer, 1991),

Page 340 and following .... 09 Jul 1995

In the next edition I'll adopt the notation $K_n(x_1, \ldots, x_n)$ for continuant polynomials, as in CMath §6.7, instead of writing $Q_n(x_1, \ldots, x_n)$.

Page 343 middle ________ 07 Oct 1995

line 13: **Theorem F** (G. Lamé, 1845). ⇝ **Theorem F.**
line 22: (This ⇝ This
lines 24–30: discovered.) ... Since ⇝ discovered. The result is essentially due to T. F. de Lagny [*Mém. Acad. Sci.* **11** (Paris, 1733), 363–364], who observed that Fibonacci numbers give the smallest numerator and denominator for continued fractions of a given length. He did not explicitly mention gcd calculation, however; the connection between Fibonacci numbers and Euclid's algorithm was first pointed out by É. Léger [*Correspondance math. et physique* **9** (1837), 483–485].

Shortly afterwards, P. J. É. Finck [*Traité élémentaire d'arithmétique* (Strasbourg, 1841), 44] proved by another method that $\gcd(u, v)$ takes at most $2\lg v + 1$ steps, when $u > v > 0$; and G. Lamé [*Comptes Rendus Acad. Sci.* **19** (Paris, 1844), 867–870] improved this to $5\lceil\log_{10}(v+1)\rceil$. Full details about these pioneering studies in the analysis of algorithms appear in a interesting review by J. O. Shallit, *Historia Mathematica* **21** (1994), 401–419. A more precise estimate of the worst case is, however, a direct consequence of Theorem F:

Corollary L. *If* $0 \le v < N$*, the number of division steps required when Algorithm 4.5.2A is applied to* u *and* v *is at most* $\lceil \log_\phi(\sqrt{5}\,N) \rceil - 2$.

Proof. After step A1 we have $v > u \bmod v$. Therefore by Theorem F, the maximum number of steps, n, occurs when $v = F_{n+1}$ and $u \bmod v = F_n$. Since

▸**Page 346** line 10 ——— 22 Jun 1984

Table 3 of Appendix B $\rightsquigarrow$ Table 3 of Appendix A

Page 346 lower half of page ——— 27 Mar 1995

lines −17 and −18: K. F. Gauss ... that year $\rightsquigarrow$ C. F. Gauss, who first thought of the problem on February 5, 1799. His notebook for 1800
line −7: rigorous proof. $\rightsquigarrow$ rigorous proof. [See Gauss's *Werke*, vol. 10^1, 552–556.]

▸**Page 348** line −4 ——— 03 Nov 1981

$R_n(0) = R_n(1) = 1 \rightsquigarrow R_n(0) = R_n(1) = 0$

Page 350 line 5 ——— 21 May 1984

65860 ... $\rightsquigarrow$ 65859 74481 21901 55623 31109−

Page 350 lower middle ——— 07 Oct 1995

lines −13 and −12: $\lambda_3 = 0.1009, \ldots$ Babenko $\rightsquigarrow$ $\lambda_3 \approx 0.10088$, $\lambda_4 \approx -0.03550$, $\lambda_5 \approx 0.01284$, $\lambda_6 \approx -0.00472$, $\lambda_7 \approx 0.00175$. Babenko
line −9: a paper by $\rightsquigarrow$ papers by
line −8: 1276.] $\rightsquigarrow$ 1276; Mayer and Roepstorff, *J. Statistical Physics* **47** (1987), 149–171; **50** (1988), 331–344; Daudé, Flajolet, and Vallée, *Lecture Notes in Comp. Sci.* **877** (1994), 144–158.] The 40-place value of λ in (39) was computed by John Hershberger.

▸**Page 351** line −9 ——— 22 Jun 1984

$n\mu(\mathcal{I}_N) - N \le s_n \le n\mu(\mathcal{J}_N) + N \rightsquigarrow n\mu(\mathcal{J}_N) - N \le s_n \le n\mu(\mathcal{J}_N) + N$

▸**Page 352** line 5 ——— 24 Jan 1983

distribution function (21). $\rightsquigarrow$ *distribution function* (22).

▸**Page 353** line 4 ——— 04 Feb 1981

partial fractions $\rightsquigarrow$ continued fractions

▸**Page 353** middle ——— 19 Jun 1997

[The '$n =$' and '$T_n =$' are too high; a similar error is on page 354.]

▸**Page 356** line 2 ——— 30 Sep 1982

$$\ln 100 - \frac{\ln 2}{2} - \frac{\ln 2}{4} - \frac{\ln 2}{5} - \frac{\ln 5}{25} \rightsquigarrow \ln 100 - \frac{\ln 2}{2} - \frac{\ln 2}{4} - \frac{\ln 5}{5} - \frac{\ln 5}{25}$$

▸**Page 356** line 9 ——— 22 Jun 1984

exercise 27 $\rightsquigarrow$ exercise 29

▸**Page 357** line 4 ——— 11 Jul 1985

$4\gamma - 24\pi^2\zeta'(2)$ ⇝ $4\gamma - 24\pi^{-2}\zeta'(2)$

Page 357 line 6 ——— 07 Oct 1995

proved. ⇝ proved. Using (58), Graham H. Norton [*J. Symbolic Computation* **10** (1990), 53–58] extended the calculations of exercise 29 to prove that the empirical constant 0.06 in (57) is actually

$$\bigl((6\ln 2)/\pi^2\bigr)\bigl(3\ln 2 + 4\gamma - 12\pi^{-2}\zeta'(2) - 3\bigr) - 1 = 0.0653514259\ldots.$$

▸**Page 357** line 13 ——— 25 Feb 1983

when $v = n$ will never ⇝ when $0 \le v < N$ will never

Page 359 line 1 ——— 08 Jan 1997

(J. Lagrange.) ⇝

▸**Page 359,** line 1 of exercise 13 ——— 11 Jan 1996

1797 ⇝ 1767

Page 361 exercise 30 ——— 11 Oct 1995

variant ⇝ “greedy” variant

▸**Page 363** line 6 ——— 18 Jun 1997

$\lfloor\log_\phi(\sqrt{5}n+1)\rfloor$ ⇝ $\lceil\log_\phi(\sqrt{5}n+1)\rceil$

Page 363 line 1 of exercise 40 ——— 29 Aug 1996

(*The Stern–Peirce tree.*) ⇝ (*The Stern–Brocot tree.*)

Page 364 replacement for exercise 45 ——— 11 Oct 1995

45. [*M21*] Let $T_0 = (1,0,u)$, $T_1 = (0,1,v)$, …, $T_{n+1} = \bigl((-1)^{n+1}v/d, (-1)^n u/d, 0\bigr)$ be the sequence of vectors computed by Algorithm 4.5.2X (the extended Euclidean algorithm), and let $/\!/a_1,\ldots,a_n/\!/$ be the regular continued fraction for v/u. Express T_j in terms of continuants involving a_1, …, a_n, for $1 < j \le n$.

46. [*M33*] By adjusting the final iteration of Algorithm 4.5.2X so that a_n is optionally replaced by two partial quotients $(a_n - 1, 1)$, we can assume that the number of iterations, n, has a given parity. Continuing the previous exercise, let λ and μ be arbitrary positive real numbers and let $\theta = \sqrt{\lambda\mu v/d}$, where $d = \gcd(u,v)$. Prove that if n is even, and if $T_j = (x_j, y_j, z_j)$, we have $\min_{j=1}^{n+1}\bigl|\lambda x_j + \mu z_j - [j \text{ even}]\,\theta\bigr| \le \theta$.

Page 364 new exercise ——— 29 Aug 1996

▸**47.** [*M30*] (*Rational reconstruction.*) The number 28481 turns out to be equal to 41/316 (modulo 199999), in the sense that $316 \cdot 28481 \equiv 41$. How could a person discover this? Given integers a and m with $m > 2$, explain how to find integers x and y such that $ax \equiv y$ (modulo m), $x \perp y$, $0 < x \le \sqrt{m/2}$, and $|y| \le \sqrt{m/2}$, or to determine that no such x and y exist. Can there be more than one solution?

Page 364 line 18 ——— 20 Apr 1996

them. ⇝ them. [A comprehensive history of factoring before 1950 has been prepared by H. C. Williams and J. O. Shallit, *Proc. Symp. Applied Math.* **48** (1993), 481–531.]

Page 366 replacement for (5) ———— 24 Jan 1983

$$\pi(x) = \sum_{k=1}^{\lg x} \frac{\mu(k)}{k} L\big(\sqrt[k]{x}\,\big) + O(1) = L(x) - \frac{1}{2}L\big(\sqrt{x}\,\big) - \frac{1}{3}L\big(\sqrt[3]{x}\,\big) + \cdots + O(1) \qquad (5)$$

Page 366 replacement for lines −11 to −14 ———— 08 Oct 1995

x	$\pi(x)$	$L(x)$	Riemann's formula
10^3	168	176.6	168.3
10^6	78498	78626.5	78527.4
10^9	50847534	50849233.9	50847455.4
10^{12}	37607912018	37607950279.8	37607910542.2
10^{15}	29844570422669	29844571475286.5	29844570495886.9
10^{18}	24739954287740860	24739954309690414.0	24739954284239494.4

[See Lagarias, Miller, and Odlyzko, *Math. Comp.* **44** (1985), 537–560; Deléglise and Rivat, *Math. Comp.* **65** (1996), 235–245.] However,

▸**Page 368** line 1 before (7) ———— 26 Jan 1986

random integer x $\leadsto$ random integer $\le x$

Page 369 lines 1 and 2 ———— 24 Feb 1995

4th ed. (Oxford, 1960), $\leadsto$

Page 369 new paragraph to follow line 10 ———— 06 Dec 1983

An excellent exposition of the probability distribution of the prime factors of a random integer has been given by Patrick Billingsley, *AMM* **80** (1973), 1099–1115; see also *Annals of Probability* **2** (1974), 749–791.

Page 369 line 11 ———— 21 Oct 1995

Factoring à la Monte Carlo. $\leadsto$ **Factoring by pseudorandom cycles.**

Page 369 lines 1, 2, and 9 after (10) ———— 11 Nov 1985

$l(m)$ $\leadsto$ $\ell(m)$ (4 changes)

Page 369 line −9 ———— 21 Oct 1995

prime. $\leadsto$ prime. Pollard called his technique the "rho method," because the eventually periodic sequence y_0, y_1, ... reminded him of the Greek letter ρ.

Page 370 line 1 ———— 21 Oct 1995

(*Monte Carlo factorization*) $\leadsto$ (*Factoring by the rho method*)

▸**Page 370** line 6 ———— 21 Jun 1997

$A = 1$ $\leadsto$ $A = 2$

Page 371 top of page ———— 16 Jun 1982

line 6: in 1980 $\leadsto$
line 7: $2^{256}+1$. $\leadsto$ $2^{256}+1$. [*Math. Comp.* **36** (1981), 627–630; **38** (1982), 253–255.]

Page 371 line 12 ________ 30 Oct 1995

(New York: Chelsea, 1952) $\leadsto$ (Carnegie Inst. of Washington, 1919)

Page 374 lines −10 to −5 ________ 27 Feb 1996

Special sieve ... is described $\leadsto$ F. W. Lawrence proposed the construction of special sieve machines for factorization in the 19th century [*Quart. J. of Pure and Applied Math.* **28** (1896), 285–311], and E. O. Carissan completed such a device with 14 moduli in 1919. [See Shallit, Williams, and Morain, *Math. Intelligencer* **17**, 3 (1995), 41–47, for the interesting story of how Carissan's long-lost sieve was rediscovered and preserved for posterity.] D. H. Lehmer and his associates constructed and used many different sieve devices during the period 1926–1989, beginning with bicycle chains and later using photoelectric cells and other kinds of technology; see, for example, *AMM* **40** (1933), 401–406. Lehmer's electronic delay-line sieve, which began operating in 1965, processed one million numbers per second. By 1995 it was possible to construct a machine that sieved 6144 million numbers per second, performing 256 iterations of steps D2 and D3 in about 5.2 nanoseconds [see Lukes, Patterson, and Williams, *Nieuw Archief voor Wiskunde* (4) **13** (1995), 113–139]. Another way to factor with sieves was described

Page 376 lines 12–13 ________ 09 Oct 1995

case of some factorizations ... 383]. $\leadsto$ case of the factorization $4x^4 + 1 = (2x^2 + 2x + 1)(2x^2 - 2x + 1)$, which Euler communicated to Goldbach in 1742 [P. H. Fuss, *Correspondance Math. et Physique* **1** (1843), 145].

Page 380 top ________ 08 Oct 1995

line 4: $4(\ln n)^2 \leadsto 2(\ln n)^2$
line 5: constant 4 $\leadsto$ constant 2
lines 6–7: Peter Weinberger ... not yet published. $\leadsto$ Eric Bach, *Math. Comp.* **55** (1990), 355–380.

Page 380 line 17 ________ 16 Oct 1996

36]. $\leadsto$ 36]. B. Arazi [*Comp. J.* **37** (1994), 219–222] has observed that Algorithm P can be speeded up significantly for large n by using Montgomery's fast method for remainders (exercise 4.3.1–42).

Page 380 lines 22–23 ________ 18 Mar 1995

[See L. M. Adleman and R. S. Rumely, to appear.] $\leadsto$
[The running time in general is $(\log n)^{O(\log\log\log n)}$; see L. M. Adleman, C. Pomerance, and R. S. Rumely, *Annals of Math.* **117** (1983), 173–206.] Adleman and Ming-Deh A. Huang later found a procedure that finds rigorous proofs of primality for all prime n, with a running time that is polynomial in $\log n$ with high probability [*Lecture Notes in Math.* **1512** (1992)].

▸**Page 382** line 3 of step E5 ________ 26 Jan 1986

terminates. The $\leadsto$ terminates. (The

Page 382 line −9 ——— 17 Jan 1984

13 more times ⇝ 14 more times

Page 383 lines 1 and 2 after (21) ——— 12 Oct 1995

If we let ... assuming ⇝ If we let $\ln m$ be approximately $\frac{1}{2}\sqrt{\ln N \ln\ln N}$, we have $r \approx \sqrt{\ln N/\ln\ln N} - 1$, assuming

▸**Page 384** lines −7, −5, and −4 ——— 11 Jan 1981

N ⇝ V

Page 384 last three lines ——— 16 Jun 1981

D. R. Hickerson ... 224. ⇝ H. C. Williams, *Math. Comp.* **36** (1981), 593–601.

Page 385 lines 1–15 ——— 25 Apr 1996

[Delete these lines; I am revising them and inserting the revision on page 386.]

Page 385 line 25 ——— 25 Mar 1981

Dixon's method ⇝ Dixon's method [*Math. Comp.* **36** (1981), 255–260]

▸**Page 386** replacement for lines 4–6 ——— 12 Oct 1995

so we can estimate the running time almost as we did in (21) but with the quantity $2\sqrt{N}$ replaced by N. This time we choose $r = \sqrt{2\ln N/\ln\ln N+\theta}$ where $|\theta| \le 1$ and r is even, and we choose m so that $r = \ln N/\ln p_m + O(1/\log\log N)$; this means

$$\begin{aligned}\ln p_m &= \sqrt{\frac{\ln N \ln\ln N}{2}} - \frac{\theta}{2}\ln\ln N + O(1),\\ \ln m &= \ln \pi(p_m) = \ln p_m - \ln\ln p_m + O(1/\log p_m)\\ &= \sqrt{\frac{\ln N \ln\ln N}{2}} - \frac{\theta+1}{2}\ln\ln N + O(\log\log\log N),\end{aligned}$$

Page 386 new material to follow Theorem D ——— 25 Apr 1996

Other approaches. Another factorization technique was suggested by John M. Pollard [*Proc. Cambridge Phil. Soc.* **76** (1974), 521–528], who gave a practical way to discover prime factors p of N when $p-1$ has no large prime factors. The latter algorithm (see exercise 19) is probably the first thing to try after Algorithms A and B have run too long on a large N.

A survey paper by R. K. Guy, written in collaboration with J. H. Conway, *Congressus Numerantium* **16** (1976), 49–89, gave a unique perspective on the developments up till that time. Guy stated, "I shall be surprised if anyone regularly factors numbers of size 10^{80} without special form during the present century"; and he was indeed destined to be surprised many times during the next 20 years.

Tremendous advances in factorization techniques for large numbers were made during the 1980s, beginning with Carl Pomerance's *quadratic sieve method* of 1981 [see *Lecture Notes in Comp. Sci.* **209** (1985), 169–182]. Then Hendrik

Lenstra devised the *elliptic curve method* [*Annals of Math.* **126** (1987), 649–673], which heuristically is expected to take about $\exp\bigl(\sqrt{(2+\epsilon)(\ln p)(\ln\ln p)}\,\bigr)$ multiplications to find a prime factor p. This is about the same running time as our estimate for Algorithm E when $p \approx \sqrt{N}$, but it is much better when N has relatively small prime factors.

John Pollard came back in 1988 with another new technique, which has become known as the *number field sieve*; see *Lecture Notes in Math.* **1554** (1993) for a series of papers about this method, which is the current champion for factoring extremely large integers. Its running time is predicted to be of order

$$\exp\bigl((64/9+\epsilon)^{1/3}(\ln N)^{1/3}(\ln\ln N)^{2/3}\bigr)$$

as $N \to \infty$. The crossover point at which a well-tuned version of the number field sieve begins to beat a well-tuned version of the quadratic sieve appears to be at $N \approx 10^{112}$, according to A. K. Lenstra.

Details of the new methods are beyond the scope of this book, but we can get an idea of their effectiveness by noting some of the early success stories in which unfactored Fermat numbers of the form $2^{2^k}+1$ were cracked. For example, the factorization

$$2^{512}+1 = 2424833\cdot\\ 7455602825647884208337395736200454918783366342657\cdot p_{99}$$

was found by the number field sieve after four months of computation by using otherwise idle time on about 700 workstations [Lenstra, Lenstra, Manasse, and Pollard, *Math. Comp.* **61** (1993), 319–349; **64** (1995), 1357]; here p_{99} denotes a 99-digit prime number. The next Fermat number has twice as many digits, but it yielded to the elliptic curve method on October 20, 1995:

$$2^{1024}+1 = 45592577\cdot 6487031809\cdot\\ 4659775785220018543264560743076778192897\cdot p_{252}.$$

[Richard Brent, to appear.] In fact, Brent had already used the elliptic curve method to resolve the next case as early as 1988:

$$2^{2048}+1 = 319489\cdot 974849\cdot\\ 167988556341760475137\cdot 3560841906445833920513\cdot p_{564};$$

by a stroke of good luck, all but one of the prime factors was $< 10^{22}$, so the elliptic curve method was a winner.

What about $2^{4096}+1$? At present, that number seems completely out of reach. It has five factors $< 10^{16}$, but the unfactored residual has 1187 decimal digits. The next case, $2^{8192}+1$, has four known factors $< 10^{27}$ [Crandall and Fagin, *Math. Comp.* **62** (1994), 321; Brent, Crandall, and Dilcher, to appear] and a huge unfactored residual.

Page 386 line -11 ——————————————— 17 Feb 1981

1979 $\leadsto$ 1978

Page 387 new paragraph to precede line −11 ——— 20 Sep 1995

Some precautions are necessary, however. If the same message is sent to three different people on a computer network, a person who knows x^3 modulo N_1, N_2, and N_3 could reconstruct $x^3 \bmod N_1N_2N_3 = x^3$ by the Chinese remainder theorem, so x would no longer be a secret. In fact, even if a "time-stamped" message $(2^{\lceil \lg t_i \rceil}x + t_i)^3 \bmod N_i$ is sent to seven different people, with known or guessable t_i, the value of x can be deduced (see exercise 44). Therefore some cryptographers have recommended encoding with the exponent $2^{16} + 1 = 65537$ instead of 3; this exponent is prime, and the computation of $x^{65537} \bmod N$ takes only about 8.5 times as long as the computation of $x^3 \bmod N$. [*CCITT Recommendations Blue Book* (Geneva: International Telecommunication Union, 1989), Fascicle VIII.8, Recommendation X.509, Annex C, pp. 74–76.] The original proposal of Rivest, Shamir, and Adleman was to encode x by $x^a \bmod N$ where a is any exponent prime to $\varphi(N)$, not just $a = 3$; in practice, however, we prefer an exponent for which encoding is faster than decoding.

▸**Page 388** line 12 ——— 12 Feb 1981

$\frac{1}{6}\ln p_1p_2 \approx 45 \leadsto \frac{1}{3}\ln p_1p_2 \approx 90$

Page 388 line −16 ——— 22 Apr 1981

644–651 $\leadsto$ 644–654

Page 388 line −4 ——— 08 Oct 1995

$D_A(M)$. $\leadsto$ $D_A(M)$. (Well, Bob himself now knows $D_A(M)$, so he could impersonate Alice by passing $E_X\bigl(D_A(M)\bigr)$ to Xavier. To defeat any such attempted forgery, the content of M should clearly indicate that it is for Bob's eyes only.)

▸**Page 389** line 20 ——— 22 Mar 1981

$\gcd(x, y) \leadsto \gcd(x - y, N)$

Page 391 updates on Mersenne ——— 15 Dec 1996

line 1: *Physico-Mathematics* $\leadsto$ *Physico-Mathematica*
line 6: correct or not. $\leadsto$ correct.
line 9: is not; therefore Mersenne was not $\leadsto$ was questionable; therefore Mersenne might not be
line 13: found $\leadsto$ showed
line 15: showed $\leadsto$ found
line 18: the first 27 $\leadsto$
lines 19–29: replace '2, 3, ... An efficient' by the following:

$$\begin{gathered}2, 3, 5, 7, 13, 17, 19, 31, 61, 89, 107, 127, 521, 607, 1279, 2203,\\ 2281, 3217, 4253, 4423, 9689, 9941, 11213, 19937, 21701, 23209, 44497,\\ 86243, 110503, 132049, 216091, 756839, 859433, 1257787, 1398269.\end{gathered} \tag{23}$$

Most of the entries on the bottom line were found by David Slowinski and associates while testing new supercomputers [see *J. Recreational Math.* **11** (1979), 258–261]; he found 756839, 859433, and 1257787 in collaboration with Paul Gage.

However, the next (and currently largest) example, 1398269, was found by Joel Armengaud with an off-the-shelf personal computer; Armengaud used a program by George Woltman, who launched the Great Internet Mersenne Prime Search project (GIMPS) in 1996. Notice that the prime $8191 = 2^{13} - 1$ does not occur in (23); Mersenne had stated that $2^{8191} - 1$ is prime, and others had conjectured that any Mersenne prime could perhaps be used in the exponent.

The search for large primes has not been systematic, because people have generally tried to set a hard-to-beat world record instead of spending time with smaller exponents. For example, $2^{132049} - 1$ was proved prime in 1983 and $2^{216091} - 1$ in 1984, but the case $2^{110503} - 1$ was not discovered until 1988. Therefore one or more unknown Mersenne primes less than $2^{1398269} - 1$ might still exist. (According to Woltman, all exponents to 750,000 have been checked, and his volunteers are systematically filling the remaining gaps.)

Since $2^{1257787} - 1$ and $2^{1398269} - 1$ are numbers of about 400,000 decimal digits each, it is clear that some special techniques have been used to prove that they are prime. (In fact, the initial verification of $2^{1257787} - 1$ on April 12, 1996, took less than 8.3 hours on a Cray T94. The initial verification of $2^{1398269} - 1$ on November 13, 1996, took about 88 hours on a 90 MHz Pentium PC.) An efficient

Page 391 replacement for the bottom line ——— 01 Jan 1997

when necessary, since calculation mod $(2^q - 1)$ is so convenient; see Section 4.3.2. Exercise 4.3.2–14 explains how to save time when q is extremely large.

Page 394 vicinity of line 10 ——— 17 Jul 1981

line 9: always $\leadsto$ almost always
line 11: will probably change soon $\leadsto$ might change

Page 394 line 10 ——— 23 Dec 1996

at least from 1772 until 1980 when this book was written. $\leadsto$ ever since 1460, by which time an anonymous author whose works are now preserved in Italian libraries had shown that $2^{17} - 1$ and $2^{19} - 1$ are prime [see E. Picutti, *Historia Math.* **16** (1989), 123–136].

Page 394 line 2 of exercise 4 ——— 11 Nov 1985

$X_{l(n)-1} \leadsto X_{\ell(n)-1}$

▸**Page 394** line 3 of exercise 8 ——— 08 Jan 1982

odd numbers less than N $\leadsto$ odd numbers between 1 and N

▸**Page 395** line 4 of exercise 13 ——— 01 Aug 1996

be found, when n $\leadsto$ not be found, when N

Page 396 line 2 ——— 31 Mar 1981

all primes $\leadsto$ all odd primes

Page 396 new sentence to follow line 3 of exercise 23 ——— 19 Apr 1995

Thus it generalizes the Legendre symbol of exercise 1.2.4–47.

▸**Page 396** exercise 24 line 2 ——— 17 Jan 1981

$x = n \leadsto x \bmod n = 0$

▸**Page 396** lines 2 and 3 of exercise 25 ——— 24 Oct 1995

$$\pi(x) + \frac{\pi(x^{1/2})}{2} + \frac{\pi(x^{1/3})}{3} + \cdots = \int_2^x \frac{dt}{\ln t} - 2\sum \int_{-\infty}^{\sigma} \frac{e^{(t+i\tau)\ln x}\,dt}{t+i\tau} + O(1)\,,$$

where the sum is over all complex $\sigma + i\tau$ such that $\tau > 0$ and $\zeta(\sigma + i\tau) = 0$.

Page 397 line 1 of exercise 31 ——— 16 May 1996

formula 3.5–33 $\leadsto$ exercise 1.2.10–22

Page 397 line 4 of exercise 34 ——— 19 Apr 1995

not too large. $\leadsto$ not too large (say $m < N^{10}$).

Page 398 new exercise ——— 05 Apr 1981

39. [*HM30*] (L. Adleman.) Let p be a rather large prime number and let a be a primitive root modulo p; thus, all integers b in the range $1 \le b < p$ can be written $b = a^n \bmod p$, for some unique n with $1 \le n < p$.

Design an algorithm that almost always finds n, given b, in $O(p^\epsilon)$ steps for all $\epsilon > 0$, using ideas similar to those of Dixon's factoring algorithm. [*Hint:* Start by building a repertoire of numbers n_i such that $a^{n_i} \bmod p$ has only small prime factors.]

Page 398 new exercise ——— 19 Apr 1995

40. [*25*] What is the largest 50-digit prime number with the maximum possible number of (a) 0s, (b) 1s, (c) 2s, (d) 3s, (e) 4s, (f) 5s, (g) 6s, (h) 7s, (i) 8s, (j) 9s, in decimal notation?

Page 398 new exercise ——— 10 Sep 1995

41. [*40*] Many primes p have the property that $2p+1$ is also prime; for example, $5 \to 11 \to 23 \to 47$. More generally, say that q is a *successor* of p if p and q are both prime and $q = 2^k p + 1$ for some $k \ge 0$. For example, $2 \to 3 \to 7 \to 29 \to 59 \to 1889 \to 3779 \to 7559 \to 4058207223809 \to 32465657790473 \to 446204603050269297187225 7 \to$ $95\langle\, 30$ omitted digits$\,\rangle 37 \to \cdots$; the smallest successor of $95\ldots37$ has 103 digits.

Find the longest chain of successive primes that you can.

Page 398 new exercise ——— 11 Oct 1995

42. [*M35*] (H. W. Lenstra, Jr.) Given $0 < r < s < N$ with $r \perp s$ and $N \perp s$, show that it is possible to find all divisors of N that are $\equiv r$ (modulo s) by performing $O\bigl(\lceil N/s^3\rceil^{1/2}\log s\bigr)$ well-chosen arithmetic operations on $(\lg N)$-bit numbers. [*Hint:* Apply exercise 4.5.3–46.]

Page 398 new exercise ——— 09 Jan 1997

▸**43.** [*M41*] Let $m = pq$ be an r-bit Blum integer as in Theorem 3.5P, and let $Q_m = \{y \mid y = x^2 \bmod m \text{ for some } x\}$. Then Q_m has $(p+1)(q+1)/4$ elements, and every element $y \in S_m$ has a unique square root $x = \sqrt{y}$ such that $x \in Q_m$. Suppose $G(y)$ is an algorithm that correctly guesses $\sqrt{y} \bmod 2$ with probability $\ge \frac12 + \epsilon$, when y is a random element of Q_m. The goal of this exercise is to prove that the problem solved by G is almost as hard as the problem of factoring m.

a) Construct an algorithm $A(G, m, \epsilon, y, \delta)$ that uses random numbers and algorithm G to guess whether a given integer y is in Q_m, without necessarily computing $\sqrt{y}$. Your algorithm should guess correctly with probability $\geq 1 - \delta$, and its running time $T(A)$ should be at most $O\big(\epsilon^{-2}(\log\delta^{-1})T(G)\big)$, assuming that $T(G) \geq r^2$. (If $T(G) < r^2$, replace $T(G)$ by $(T(G) + r^2)$ in this formula.)

b) Construct an algorithm $F(G, m, \epsilon)$ that finds the factors of m with expected running time $T(F) = O\big(r^2(\epsilon^{-6} + \epsilon^{-4}(\log\epsilon^{-1})T(G))\big)$.

Hints: For fixed $y \in Q_m$, and for $0 \leq v < m$, let $\tau v = v\sqrt{y} \bmod m$ and $\lambda v = \tau v \bmod 2$. Notice that $\lambda(-v) + \lambda v = 1$ and $\lambda(v_1 + \cdots + v_n) = \big(\lambda v_1 + \cdots + \lambda v_n + \lfloor(\tau v_1 + \cdots + \tau v_m)/m\rfloor\big) \bmod 2$. Furthermore we have $\tau(\frac{1}{2}v) = \frac{1}{2}(\tau v + m\lambda v)$; here $\frac{1}{2}v$ stands for $(\frac{m+1}{2}v) \bmod m$. If $\pm v \in Q_m$ we have $\tau(\pm v) = \sqrt{v^2 y}$; therefore algorithm G gives us a way to guess λv for about half of all v.

Page 398 new exercise ________ 20 Oct 1995

44. [*M35*] (J. Håstad.) Show that it is not difficult to find x when $a_{i0} + a_{i1}x + a_{i2}x^2 + a_{i3}x^3 \equiv 0$ (modulo m_i), $0 < x < m_i$, $\gcd(a_{i0}, a_{i1}, a_{i2}, a_{i3}, m_i) = 1$, and $m_i > 10^{27}$ for $1 \leq i \leq 7$, if $m_i \perp m_j$ for $1 \leq i < j \leq 7$. (All variables are integers; all but x are known.) *Hint:* When L is any nonsingular matrix of real numbers, the algorithm of Lenstra, Lenstra, and Lovász [*Mathematische Annalen* **261** (1982), 515–534] efficiently finds a nonzero integer vector $v = (v_1, \ldots, v_n)$ such that $\mathrm{length}(vL) \leq \sqrt{n2^n}\,|\det L|^{1/n}$.

Page 398 new exercise ________ 21 Aug 1996

▸**45.** [*M41*] (J. M. Pollard and Claus-Peter Schnorr.) Show that there's an efficient way to solve the congruence

$$x^2 - ay^2 \equiv b \pmod{n}$$

for integers x and y, given integers a, b, and n with $ab \perp n$ and n odd, even if the factorization of n is unknown. [*Hint:* Use the identity $(x_1^2 - ay_1^2)(x_2^2 - ay_2^2) = x^2 - ay^2$, where $x = x_1x_2 - ay_1y_2$ and $y = x_1y_2 + x_2y_1$.]

Page 398 new exercise ________ 20 Oct 1995

46. [*M50*] A certain literary quotation $x = x_1x_2$, represented in ASCII code, has the enciphered value $(x_1^3 \bmod N, x_2^3 \bmod N) =$

```
(14E97EF5C531D92591B89CDBAB48444A04612C01AA29C2A8FA10FA804EF7AC3CE03D7D3667C4D3E132A24A68
 E6797FE28650DC3ADF327474B86B0CBD5387A49872CE012269A59B3E4B3BD83B74681A78AD7B6D1772A7451B,
 15B025E2AEE095A9542590184CF62F72B2E8E8DD794AEF8511F2591E6BC2C8B8A8E48AF1FE04FF2FD933E730
 9205A3418DBB9BB8C6A7665DA309531735FE86C741D1261B34CB2668FA34D0C0C28575A2454E3DB00E408AC7)
```

in hexadecimal notation, where N is

```
17B2353B9595ECA69FEF80940160C4084286D1255FFE49D114F2E633F82C88D5224FC4AA6F9104CED2BCA810
BEA76157FFDC78F9656A0ED9B3F6CCAB99001B8B2571F4EBD095925F07F9BEE5111E8375DFD71593628AD8D1.
```

What is x?

▸**Page 402** line 15 ________ 15 Feb 1981

$r_1(x) = 0.$ $\rightsquigarrow$ $r_1(x) = r_2(x).$

Page 405 line 9 ________ 13 Jul 1996

depending on u and v $\rightsquigarrow$ depending on the way contents are calculated

▸**Page 407** line −2 ——— 03 Mar 1981

$\gcd(v(x), \mathrm{pp}((r(x)))$ ⇝ $\gcd(v(x), \mathrm{pp}(r(x)))$

Page 409 lines −10 to −6 ——— 24 Aug 1996

will run noticeably slower ... with (6) above, ⇝ tends to run slower than the all-integer Algorithm E, since rational arithmetic usually requires more evaluations of integer gcds within each step when the polynomials have large degree.

It is instructive to compare (12), (13), and (14) with (6) above,

▸**Page 414** line −4 ——— 05 Jun 1981

(25) ⇝ (26)

▸**Page 415** line 7 ——— 05 Jun 1981

(16) and (17) ⇝ (17) and (18)

Page 416 and previous pages . . . . 21 Oct 1995

In the next edition I plan to redo the exposition of this section, taking into proper account the 19th-century research on determinants, as well as the work of W. Habicht, Comm. Math. Helvetici **21** *(1948), 99–116. An excellent discussion of the latter has been given by R. Loos, Computing, Supplement 4 (1982), 115–137. An interesting method for evaluating determinants, derived by C. L. Dodgson (aka Lewis Carroll) from a theorem of Jacobi, is also highly relevant to these methods. See D. E. Knuth, Electronic J. Combinatorics* **3***, 2 (1996), paper R5, §3, for a summary of the early history of identities between determinants of submatrices.*

Page 416 line 7 ——— 24 Aug 1996

divisors; another ⇝ divisors and resultants. Another

▸**Page 416** line 9 ——— 13 Jul 1996

présentes ⇝ *présentés*

Page 418 replacement for exercise 16 ——— 20 Apr 1995

▸**16.** [*M22*] Let $f(x_1, \ldots, x_n)$ be a multivariate polynomial that is not identically zero, and let $r(S_1, \ldots, S_n)$ be the set of roots $(x_1, \ldots, x_n)$ of $f(x_1, \ldots, x_n) = 0$ such that $x_1 \in S_1$, ..., $x_n \in S_n$. If the degree of f is at most $d_j \le |S_j|$ in the variable x_j, prove that

$$\bigl|r(S_1, \ldots, S_n)\bigr| \;\le\; |S_1| \ldots |S_n| - \bigl(|S_1| - d_1\bigr) \ldots \bigl(|S_n| - d_n\bigr)\,.$$

Therefore the probability of finding a root at random, $|r(S_1, \ldots, S_n)|/|S_1| \ldots |S_n|$, approaches zero as the sets S_j get bigger. [This inequality has many applications in the design of randomized algorithms, because it provides a good way to test whether a complicated sum of products of sums is identically zero without expanding out all the terms.]

Page 420 new exercise ——— 25 Aug 1996

27. [*M23*] Suggest a way to divide $u(x)$ by $v(x)$ when we know that $u(x)$ is exactly divisible by $v(x)$.

Page 421 line 4 after (4) ——— 30 May 1991

(see exercise 34) $\rightsquigarrow$ (see exercises 34 and 36)

▸**Page 428** line −15 ——— 26 Jun 1997

completed in $O\bigl(n^2(\log p)^3\bigr)$ $\rightsquigarrow$ completed in $O\bigl(n^3(\log p)^2\bigr)$

▸**Page 429** line −19 ——— 14 Sep 1996

$O\bigl(r(\log p)^2\bigr)$ steps $\rightsquigarrow$ $O\bigl(rn(\log p)^2\bigr)$ steps

Page 429 lines −15 and −14 ——— 22 Oct 1995

[Delete this paragraph.]

Page 429 line −5 ——— 02 Feb 1981

$c < d \rightsquigarrow 1 \le c < d$

Page 430 lines 13–19 ——— 22 Oct 1995

[Delete this paragraph.]

▸**Page 430** line −10 ——— 22 Feb 1981

$\gcd\bigl(g_d(x), t(x)^{(p^d-1)/2}\bigr) \rightsquigarrow \gcd\bigl(g_d(x), t(x)^{(p^d-1)/2} - 1\bigr)$

Page 430 line −4 ——— 22 Oct 1995

[Delete this line.]

Page 431 line 15 ——— 14 Sep 1996

exercise 30. $\rightsquigarrow$ exercise 30. A faster algorithm for distinct-degree factorization when p is very large has been found by E. Kaltofen and V. Shoup; see *J. Symbolic Comp.* **20** (1995), 363–397.

Page 431 new paragraph to follow line 15 ——— 22 Oct 1995

Historical notes: The idea of finding all the linear factors of a squarefree polynomial $f(x)$ modulo p by first calculating $g(x) = \gcd\bigl(x^{p-1} - 1, f(x)\bigr)$ and then calculating $\gcd\bigl(g(x), (x+s)^{(p-1)/2} \pm 1\bigr)$ for arbitrary s is due to A. M. Legendre, *Mémoires Acad. Sci.* (Paris, 1785), 484–490; his motive was to find all of the integer solutions to Diophantine equations of the form $f(x) = py$, i.e., $f(x) \equiv 0$ (modulo p). The more general degree-separation technique embodied in Algorithm D was discovered by C. F. Gauss before 1800, but not published [see his *Werke* **2** (1876), 237], and then by Évariste Galois in the now-classic paper that launched the theory of finite fields [*Bulletin des Sciences Mathématiques, Physiques et Chimiques* **13** (1830), 428–435; reprinted in *J. de Math. Pures et Appliquées* **11** (1846), 398–407]. However, this work of Gauss and Galois was ahead of its time, and not well understood until J. A. Serret gave a detailed exposition somewhat later [*Mémoires Acad. Sci.*, series 2, **35** (Paris, 1866), 617–688; Algorithm D is in §7]. Special procedures for splitting $g_d(x)$ into irreducible factors were devised subsequently by various authors, but methods of full generality that would work efficiently for large p were apparently not discovered until the advent of computers made them desirable. The first such randomized algorithm with a rigorously analyzed running time was published by E. Berlekamp [*Math. Comp.* **24** (1970), 713–735]; it was refined and simplified by Robert T. Moenck [*Math. Comp.* **31** (1977), 235–250], M. O. Rabin [*SICOMP* **9** (1980), 273–280], D. G. Cantor and H. J. Zassenhaus [*Math. Comp.* **36** (1981), 587–592]. Paul Camion independently found a generalization to special classes of multivariate polynomials [*Comptes Rendus* **A291** (1980), 479–482; *IEEE Trans.* **IT-29** (1983), 378–385].

Page 431 new paragraph to precede line 16 ———— 14 Sep 1996

The average number of operations needed to factor a random polynomial mod p has been analyzed by P. Flajolet, X. Gourdon, and D. Panario, *Lecture Notes in Comp. Sci.* **1099** (1996), 232–243.

▸**Page 432** line −9 ———— 11 Jun 1981

$(x^2 - 13 - 7) \leadsto (x^2 - 13x - 7)$

▸**Page 432** line −5 ———— 30 Sep 1982

$-12 \times 7 \leadsto 12 \times (-7)$

Page 433 lines 7–10 ———— 27 Oct 1995

introduced by K. Hensel ... if we $\leadsto$ known as Hensel's Lemma can be used to extend a factorization modulo p in a unique way to a factorization modulo p^e for arbitrarily high exponents e (see exercise 22). If we

Page 433 line −14 ———— 29 Aug 1996

instead of $u(x)$. $\leadsto$ instead of $u(x)$. Another way to solve the leading coefficient problem is discussed in exercise 40.

▸**Page 433** bottom line ———— 21 May 1981

$d > \frac{1}{2}r \leadsto d \le \frac{1}{2}r$

Page 434 new paragraph to follow line 9 ———— 29 Aug 1996

We can decrease B still more if we decide to guarantee only that B should bound the coefficients of *at least one* proper divisor of $u(x)$. (For example, when we're factoring a non-prime integer N instead of a polynomial, some of the divisors might be very large, but at least one will be $\le \sqrt{N}$.) This idea, due to B. Beauzamy, V. Trevisan, and P. S. Wang [*J. Symbolic Comp.* **15** (1993), 393–413], is discussed in exercise 21. The divisibility test in step F2 must then be applied to both $v(x)$ and $w(x)$, but the computations are faster because p^e is often much smaller.

Page 434 line 11 ———— 27 Apr 1981

$2^{r-1} \leadsto 2^{r-1} - 1$

▸**Page 434** line 24 ———— 24 Mar 1983

$i_1 + \cdots + i_d \in \bigcap D_j \leadsto \deg(i_1) + \cdots + \deg(i_d) \in \bigcap D_j$

Page 434 new material to precede line −9 ———— 01 Nov 1995

A. K. Lenstra, H. W. Lenstra, and L. Lovász introduced their famous "LLL algorithm" in order to obtain rigorous worst-case bounds on the amount of computation needed to factor a polynomial over the integers [*Math. Annalen* **261** (1982), 515–534]. Their method requires no random numbers, and its running time for $u(x)$ of degree n is $O\bigl(n^{12} + n^9(\log \|u\|)^3\bigr)$ bit operations, where $\|u\|$ is defined in exercise 20. This estimate includes the time to search for a suitable prime number p and to find all factors modulo p with Algorithm B. Of course, heuristic methods that use randomization run noticeably faster in practice.

Page 436 in the next edition . . . . 21 Apr 1995

Detailed algorithms for multivariate polynomial computation are beyond the scope of my book, but eventually I'll want to give at least a reference to the current state of the art. For example, the methods currently cited on this page are inferior (at least in some important cases) to the randomized algorithm of Zippel cited in the new answer to exercise 4.6.1–16; and significant computational experience has been gained since the days when those pioneering papers were written.

Page 436 new quotation to precede the exercises ———— 22 Oct 1995

The asymptotically best algorithms frequently turn out
to be worst on all problems for which they are used.
— D. G. CANTOR and H. ZASSENHAUS (1981)

▸**Page 437** exercise 5 ———— 14 Sep 1996

factors $\rightsquigarrow$ irreducible factors (twice)

Page 438 exercise 19 ———— 14 Jun 1982

by G. Eisenstein . . . 167: $\rightsquigarrow$ by T. Schönemann in *Crelle* **32** (1846), 100, then popularized by G. Eisenstein in *Crelle* **39** (1850), 166–169:

Page 438 exercise 20 ———— 24 Apr 1995

$|u| \rightsquigarrow \|u\|$ (in three places)
line 2: $g(x)$ and $h(x) \rightsquigarrow v(x)$ and $w(x)$ (respectively)
line 3: $|g| = |h| \rightsquigarrow \|v\| = \|w\|$
lines 5 and 6: $M(u) = \cdots j \le n.$ $\rightsquigarrow$ $M(u) = |u_n| \prod_{j=1}^{n} \max(1, |\alpha_j|)$. Prove that $M(u) \le \|u\|$. (c) Show that $|u_j| \le \binom{n-1}{j} M(u) + \binom{n-1}{j-1}|u_n|$, for $0 \le j \le n$.

Page 438 and 439, replacement for exercise 21 ———— 29 Aug 1996

21. [*HM32*] Continuing exercise 20, we will derive useful bounds on the coefficients of *multivariate* polynomial factors over the integers. For convenience we will let boldface letters stand for sequences of t integers; thus, instead of writing

$$u(x_1, \ldots, x_t) = \sum_{j_1, \ldots, j_t} u_{j_1 \ldots j_t} x_1^{j_1} \ldots x_t^{j_t}$$

we will write simply $u(\mathbf{x}) = \sum_{\mathbf{j}} u_{\mathbf{j}} \mathbf{x}^{\mathbf{j}}$. Notice the convention for $\mathbf{x}^{\mathbf{j}}$; we also write $\mathbf{j}! = j_1! \ldots j_t!$ and $\Sigma\mathbf{j} = j_1 + \cdots + j_t$.

a) Prove the identity

$$\sum_{\mathbf{j},\mathbf{k}} \frac{1}{\mathbf{j}!\,\mathbf{k}!} \sum_{\mathbf{p},\mathbf{q}\ge 0} [\mathbf{p}-\mathbf{j}=\mathbf{q}-\mathbf{k}]\, a_{\mathbf{p}} b_{\mathbf{q}} \frac{\mathbf{p}!\,\mathbf{q}!}{(\mathbf{p}-\mathbf{j})!} \sum_{\mathbf{r},\mathbf{s}\ge 0} [\mathbf{r}-\mathbf{j}=\mathbf{s}-\mathbf{k}]\, c_{\mathbf{r}} d_{\mathbf{s}} \frac{\mathbf{r}!\,\mathbf{s}!}{(\mathbf{r}-\mathbf{j})!}$$
$$= \sum_{\mathbf{i}\ge 0} \mathbf{i}! \sum_{\mathbf{p},\mathbf{s}\ge 0} [\mathbf{p}+\mathbf{s}=\mathbf{i}]\, a_{\mathbf{p}} d_{\mathbf{s}} \sum_{\mathbf{q},\mathbf{r}\ge 0} [\mathbf{q}+\mathbf{r}=\mathbf{i}]\, b_{\mathbf{q}} c_{\mathbf{r}}\,.$$

b) The polynomial $u(\mathbf{x}) = \sum_{\mathbf{j}} u_{\mathbf{j}} \mathbf{x}^{\mathbf{j}}$ is called *homogeneous* of degree n if each term has total degree n; thus, $\Sigma\,\mathbf{j} = n$ whenever $v_{\mathbf{j}} \ne 0$. Consider the weighted sum of coefficients $B(u) = \sum_{\mathbf{j}} \mathbf{j}!\,|u_{\mathbf{j}}|^2$. Use part (a) to show that $B(u) \ge B(v)B(w)$ whenever $u(\mathbf{x}) = v(\mathbf{x})w(\mathbf{x})$ is homogeneous.

c) The *Bombieri norm* $[u]$ of a polynomial $u(\mathbf{x})$ is defined to be $\sqrt{B(u)/n!}$ when u is homogeneous of degree n. It is also defined for nonhomogeneous polynomials, by adding a new variable x_{t+1} and multiplying each term by a power of x_{t+1} so that u becomes homogeneous without increasing its maximum degree. For example, let $u(x) = 4x^3 + x - 2$; the corresponding homogeneous polynomial is $4x^3 + xy^2 - 2y^3$, and we have $[u]^2 = (3!\,0!\,4^2 + 1!\,2!\,1^2 + 0!\,3!\,2^2)/3! = 16 + \frac{1}{3} + 4$. If $u(x,y,z) = 3xy^3 - z^2$ we have $[u]^2 = (1!\,3!\,0!\,0!\,3^2 + 0!\,0!\,2!\,2!\,1^1)/4! = \frac{9}{4} + \frac{1}{6}$. What does part (b) tell us about the relation between $[u]$, $[v]$, and $[w]$, when $u(\mathbf{x}) = v(\mathbf{x})w(\mathbf{x})$?

d) Prove that if $u(x)$ is a reducible polynomial of degree n in one variable, it has a factor whose coefficients are at most $n!^{1/4}[u]^{1/2}/(n/4)!$ in absolute value. What is the corresponding result for homogeneous polynomials in t variables?

e) Calculate $[u]$ both explicitly and asymptotically when $u(x) = (x^2-1)^n$.

f) Prove that $[u][v] \ge [uv]$.

g) Show that $2^{-n/2}M(u) \le [u] \le 2^{n/2}M(u)$, when $u(x)$ is a polynomial of degree n and $M(u)$ is the quantity defined in exercise 20. (Therefore the bound in part (d) is roughly the square root of the bound we obtained in that exercise.)

Page 441 replacements for exercise 38 ——— 02 Aug 1981

38. [*HM27*] (*Perron's criterion.*) Let $u(x) = x^n + u_{n-1}x^{n-1} + \cdots + u_0$ be a polynomial with integer coefficients such that $u_0 \ne 0$ and either $|u_{n-1}| > 1 + |u_{n-2}| + \cdots + |u_0|$ or $(u_{n-1} = 0$ and $u_{n-2} > 1 + |u_{n-3}| + \cdots + |u_0|)$. Show that $u(x)$ is irreducible over the integers. [*Hint:* Prove that almost all of u's roots are less than 1 in absolute value.]

39. [*HM40*] (David G. Cantor.) Show that if the polynomial $u(x)$ is irreducible over the integers, it has a "succinct" proof of irreducibility, in the sense that the number of bits in the proof is at most a polynomial in $\deg(u)$ and the length of the coefficients. (Only a bound on the *length* of proof is requested here, as in exercise 4.5.4–17, not a bound on the time needed to find such a proof.) *Hint:* If $v(x)$ is irreducible and t is any polynomial over the integers, all factors of $v(t(x))$ have degree $\ge \deg(v)$.

Page 441 new exercises ——— 29 Aug 1996

▸**40.** [*M20*] (P. S. Wang.) If u_n is the leading coefficient of $u(x)$ and B is a bound on coefficients of some factor of u, the text's factorization algorithm requires us to find a factorization modulo p^e where $p^e > 2|u_n|B$. But $|u_n|$ might be larger than B, when B is chosen by the method of exercise 21. Show that if $u(x)$ is reducible, there is a way to recover one of its true factors from a factorization modulo p^e whenever $p^e \ge 2B^2$, by using the algorithm of exercise 4.5.3–47.

41. [*M47*] (Beauzamy, Trevisan, and Wang.) Prove or disprove: There is a constant c such that, if $f(x)$ is any integer polynomial with all coefficients $\le B$ in absolute value, then one of its irreducible factors has coefficients bounded by cB.

Page 441 bottom ——— 13 Apr 1990

line −11: Pingala's ... sûtra ⇝ Piṅgala's Hindu classic *Chandaḥ-Sûtra*
line −10: (Bombay: 1935), 76]; however, ⇝ (Lahore, 1935), 76]. However,

Page 442 replacement for lines −7 through −2 ——— 12 Nov 1987

	N	Y	Z
After step A1	23	1	x
After step A5	11	x	x^2
After step A5	5	x^3	x^4
After step A5	2	x^7	x^8
After step A5	1	x^7	x^{16}
After step A4	0	x^{23}	x^{16}

Page 443 near the top ——— 21 Apr 1995

line 4: 1800 B.C. ⇝ 2000 B.C.
line 6: This ⇝ [See A. B. Chace, *The Rhind Mathematical Papyrus* (1927); W. W. Struve, *Quellen und Studien zur Geschichte der Mathematik* **A1** (1930).] This

▸**Page 443** line −22 ——— 11 Nov 1986

an integer > 1, binary methods are no help ⇝ an integer greater than the computer word size, binary methods are little or no help

▸**Page 446** line 5 after Fig. 14 ——— 18 Jul 1997

E. C. Thurber ⇝ E. G. Thurber

Page 451 line 4 ——— 14 Jan 1985

(see exercise 29) ⇝ (see exercise 28)

▸**Page 457** line 3 ——— 28 Jun 1997

$> e_0 + d_u - d.$ ⇝ $> e_0 + d_u - d - m_0.$

Page 457 replacement for the bottom four lines ——— 23 Oct 1995

8345, 12509. The smallest n with $\nu(n) = 5$ and $l(n) \ne l^*(n)$ is $16537 = 2^{14}+9\cdot 17$ (see exercise 15).

Jan van Leeuwen has generalized Theorem H to show that

$$l^*(k2^{e_0}) + t \le l^*(kn) \le l^*(k2^{e_t}) + e_0 - e_t + t$$

for all fixed $k \ge 1$, if the exponents $e_0 > \cdots > e_t$ are far enough apart [*Crelle* **295** (1977), 202–207].

Page 458 replacement for lines 22–28 ——— 23 Oct 1995

r	$c(r)$	r	$c(r)$	r	$c(r)$	r	$c(r)$
1	2	7	29	13	607	19	18287
2	3	8	47	14	1087	20	34303
3	5	9	71	15	1903	21	65131
4	7	10	127	16	3583	22	110591
5	11	11	191	17	6271	23	196591
6	19	12	379	18	11231		

Page 458 lines 32 and 33 ——— 23 Oct 1995

[See ... growth of $c(r)$.] $\leadsto$ [The values listed here for $r > 18$ have been computed by Achim Flammenkamp. See ... growth of $c(r)$. Flammenkamp notes that $10 \cdot 2^r/r^2 < c(r) < 14 \cdot 2^r/r^2$ for all known values of $c(r)$ when $r > 5$.]

Page 459 replacement for lines 3–8 following Table 1 ——— 23 Oct 1995

r	$d(r)$	r	$d(r)$	r	$d(r)$	r	$d(r)$
1	1	6	15	11	246	16	4490
2	2	7	26	12	432	17	8170
3	3	8	44	13	772	18	14866
4	5	9	78	14	1382	19	27128
5	9	10	136	15	2481	20	49544

▸**Page 459** line −7 ——— 01 May 1997

class II $\leadsto$ Abteilung II

▸**Page 460** line 23 ——— 22 May 1986

then $a_k = a_j - b_j$ $\leadsto$ then $a_k = a_i - b_j$

▸**Page 467** in (2) ——— 13 Jul 1996

(($\leadsto$ (

Page 467 line −13 ——— 31 Mar 1987

150 years earlier. In $\leadsto$ more than 150 years earlier. For example, in

Page 467 lines −7 and −6 ——— 31 Mar 1987

[See ... 222.] $\leadsto$ Newton had been using the idea for several years in unpublished notes. [See *The Mathematical Papers of Isaac Newton*, ed. by D. T. Whiteside, **1** (1967), 490, 531; **2** (1968), 222.]

Page 467 line −6 ——— 28 Dec 1996

222.] $\leadsto$ 222.] Independently, a method equivalent to Horner's had in fact been used in 13th-century China by Ch'in Chiu Shao [see Y. Mikami, *The Development of Mathematics in China and Japan* (1913), 73–77].

▸**Page 470** step D3 ——— 04 Jul 1997

line 1: $n - j \leadsto n - 1 - j$
line 2: a multiple $\leadsto$ a positive multiple

▸**Page 474** line 6 ____ 11 Nov 1985

$m = \lfloor n/2 \rfloor - 1 \rightsquigarrow m = \lceil n/2 \rceil$

Page 475 new paragraph to follow line 10 ____ 26 Oct 1995

Questions of numerical accuracy do not arise, of course, when we are working with integers modulo m instead of with real numbers. Scheme (9) works for $n = 4$ when m is relatively prime to $2u_4$, and (16) works for $n = 6$ when m is relatively prime to $6u_6$ and to the denominator of (17). Exercise 65 shows that $n/2 + O(\log n)$ multiplications and $O(n)$ additions suffice for any monic nth degree polynomial modulo any m.

▸**Page 475** line 17 ____ 30 Jul 1996

(955) $\rightsquigarrow$ (1955)

Page 480 line 4 before (33) ____ 02 Oct 1996

411. $\rightsquigarrow$ 411. Exercise 67 discusses an interesting division-free method that involves $O(n^4)$ steps.

Page 482 updates on theoretical matrix multiplication . . . . 10 Jan 1981

Line 16 has a reference "to appear," and the paper I mentioned appears in Computers and Math. **7** *(1981), 73–125. However, in the next edition I'll be discussing the later developments instead; the best current bound is $O(n^{2.376})$, due to Coppersmith and Winograd, J. Symbolic Comp.* **9** *(1990), 251ff. This (or its sequel) will replace the existing exercises 60–64 on pages 503–505, and I'll modify the comment at the end of answer 12 on page 641.*

Page 484 bottom ____ 30 Oct 1995

line −5: Lagrange's formula $\rightsquigarrow$ formula
line −3 and −2: Unfortunately, . . . formula; $\rightsquigarrow$ Lagrange presented (41) to his class at the Paris École Normale in 1795 [see his *Œuvres* **7** (Paris, 1877), 286]; but Edward Waring of Cambridge University actually deserves the credit, because he had already presented the same formula quite clearly and explicitly in *Philosophical Transactions* **69** (1779), 59–67.

There seem to be quite a few additions, subtractions, multiplications, and divisions in Waring and Lagrange's formula;

▸**Page 484** bottom line ____ 06 May 1981

$2n^2 + 2 \rightsquigarrow 2n^2 + 2n$

Page 486 bottom ____ 27 Oct 1995

line −6: R. Moenck and A. Borodin $\rightsquigarrow$ A. Borodin and R. Moenck
line −5: 336–385; and A. Borodin $\rightsquigarrow$ 366–385; A. Borodin
line −3: 180. $\rightsquigarrow$ 180; D. Bini and V. Pan, *Polynomial and Matrix Computations* **1** (Birkhäuser, 1994), Chapter 1.]

▸**Page 488** replacement for (50) ——— 13 May 1983

$$\begin{aligned}
&\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}\begin{pmatrix}0&0&0&0\\0&0&0&0\\1&0&0&0\\0&1&0&0\end{pmatrix}\begin{pmatrix}0&0&1&0\\0&0&0&1\\0&0&0&0\\0&0&0&0\end{pmatrix}\begin{pmatrix}0&0&0&0\\0&0&0&0\\0&0&1&0\\0&0&0&1\end{pmatrix}=\begin{pmatrix}1&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}\begin{pmatrix}1&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}\begin{pmatrix}1&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}\begin{pmatrix}1&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}\\
&\quad+\begin{pmatrix}0&0&0&0\\0&1&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}\begin{pmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}\begin{pmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}\begin{pmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}+\begin{pmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}\begin{pmatrix}0&0&\bar1&1\\0&0&0&0\\0&0&1&\bar1\\0&0&0&0\end{pmatrix}\begin{pmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}\begin{pmatrix}0&0&\bar1&1\\0&0&0&0\\0&0&1&\bar1\\0&0&0&0\end{pmatrix}\\
&\quad+\begin{pmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}\begin{pmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\\bar1&1&1&\bar1\end{pmatrix}\begin{pmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}\begin{pmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}+\begin{pmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}\begin{pmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}\begin{pmatrix}0&0&0&0\\0&0&0&0\\\bar1&0&1&0\\\bar1&0&1&0\end{pmatrix}\begin{pmatrix}0&0&0&0\\0&0&0&0\\\bar1&0&1&0\\\bar1&0&1&0\end{pmatrix}\\
&\quad+\begin{pmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}\begin{pmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}\begin{pmatrix}0&0&0&1\\0&0&0&1\\0&0&0&\bar1\\0&0&0&\bar1\end{pmatrix}\begin{pmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}+\begin{pmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}\begin{pmatrix}\bar1&0&1&\bar1\\0&0&0&0\\1&0&\bar1&1\\1&0&\bar1&1\end{pmatrix}\begin{pmatrix}\bar1&0&1&\bar1\\0&0&0&0\\1&0&\bar1&1\\1&0&\bar1&1\end{pmatrix}\begin{pmatrix}\bar1&0&1&\bar1\\0&0&0&0\\1&0&\bar1&1\\1&0&\bar1&1\end{pmatrix}.
\end{aligned}\tag{50}$$

▸**Page 489** replacement for (51) ——— 13 May 1983

$$A=\begin{pmatrix}1&0&\bar1&0&0&1&\bar1\\0&1&0&0&0&1&0\\0&0&1&0&1&\bar1&1\\0&0&0&1&1&\bar1&1\end{pmatrix},\ B=\begin{pmatrix}1&0&0&\bar1&\bar1&0&1\\0&1&0&1&0&0&0\\0&0&1&1&1&0&\bar1\\0&0&\bar1&\bar1&0&1&1\end{pmatrix},\ C=\begin{pmatrix}1&1&0&0&0&0&0\\1&0&1&1&0&0&1\\1&0&0&0&1&1&1\\1&0&1&0&1&0&1\end{pmatrix}.\tag{51}$$

Page 491 lines 3 and 4 ——— 31 Oct 1995

to evaluate ... one can use the identity $\leadsto$ we can always evaluate the coefficients of $x(u)y(u) \bmod p(u)$ by using the trivial identity

Page 496 line 26 ——— 18 Oct 1996

SIAM ... 462. $\leadsto$ *SICOMP* **8** (1979), 443–462; *JACM* **27** (1980), 822–830. See also his interesting discussion of commutative bilinear forms in *SICOMP* **9** (1980), 713–728. However, the problem of computing the tensor rank of an arbitrary $n\times n\times n$ tensor over any finite field is NP-complete [J. Håstad, *Journal of Algorithms* **11** (1990), 644–654].

▸**Page 499** line 4 of exercise 25 ——— 05 Jul 1997

$\lambda_7 = \alpha_7 + \lambda_6 \leadsto \lambda_7 = \alpha_7 \times \lambda_6$

▸**Page 502** rating of exercise 48 ——— 26 Mar 1982

[*M49*] $\leadsto$ [*M21*]

▸**Page 503** last line of exercise 58 ——— 05 Jul 1997

$t = 8 \leadsto r = 8$

Page 503 last line of exercise 59 ——— 31 Oct 1995

by 2. $\leadsto$ by 2. [*Hint:* A cyclic convolution of order $2n$ can be reduced to cyclic and negacyclic convolutions of order n, using (57).]

▸**Page 505** line 1 of exercise 63 ——— 13 May 1983

exercise 59. $\leadsto$ exercise 60.

Page 505 new exercise ——— 26 Oct 1995

▸**65.** [*M25*] Show that any monic polynomial $u(x) = x^n + u_{n-1}x^{n-1} + \cdots + u_0$ can be evaluated with $\frac{1}{2}n + O(\log n)$ multiplications and $\le \frac{5}{4}n$ additions, using parameters α_1, α_2, ... that are polynomials in u_{n-1}, u_{n-2}, ... with integer coefficients. [*Hint:* Consider first the case $n = 2^l$.]

Page 505 new exercise ——— 04 Sep 1996

▸**66.** [*HM27*] (V. Strassen.) Show that the determinant (31) of an $n \times n$ matrix can be evaluated by doing $O(n^5)$ multiplications and $O(n^5)$ additions or subtractions, and no divisions. [*Hint:* Consider $\det(I + Y)$ where $Y = X - I$.]

Page 505 new exercise ——— 02 Oct 1996

67. [*HM25*] The *characteristic polynomial* $f_X(\lambda)$ of a matrix X is defined to be $\det(\lambda I - X)$. Prove that if $X = \left(\begin{smallmatrix} x & u \\ v & Y \end{smallmatrix}\right)$, where X, u, v, and Y are respectively of sizes $n \times n$, $1 \times (n-1)$, $(n-1) \times 1$, and $(n-1) \times (n-1)$, we have

$$f_X(\lambda) = f_Y(\lambda)\left(\lambda - x - \frac{uv}{\lambda} - \frac{uYv}{\lambda^2} - \frac{uY^2v}{\lambda^3} - \cdots\right).$$

Show that this relation allows us to compute the coefficients of f_X with about $\frac{1}{4}n^4$ multiplications, $\frac{1}{4}n^4$ addition/subtractions, and no divisions. *Hint:* Use the identity

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} = \begin{pmatrix} I & 0 \\ 0 & D \end{pmatrix} \begin{pmatrix} A - BD^{-1}C & B \\ 0 & I \end{pmatrix} \begin{pmatrix} I & 0 \\ D^{-1}C & I \end{pmatrix},$$

which holds for any matrices A, B, C, and D of respective sizes $l \times l$, $l \times m$, $m \times l$, and $m \times m$ when D is nonsingular.

Page 505 new exercise ——— 04 Sep 1996

▸**68.** [*HM30*] A *quolynomial chain* is like a polynomial chain except that it allows division as well as addition, subtraction, and multiplication. Prove that if $f(x_1, \ldots, x_n)$ can be computed by a quolynomial chain that has m chain multiplications and d divisions, then $f(x_1, \ldots, x_n)$ and all n of its partial derivatives $\partial f(x_1, \ldots, x_n)/\partial x_k$ for $1 \le k \le n$ can be computed by a single quolynomial chain that has at most $3m + d$ chain multiplications and $2d$ divisions. (Consequently, for example, an efficient method for calculating the determinant of a matrix leads to an efficient method for calculating all of its cofactors, hence an efficient method for computing the inverse matrix.)

Page 505 new exercise ——— 04 Jan 1997

69. [*HM35*] (A. Nozaki, 1978.) Most of the theory of polynomial evaluation is concerned with bounds on chain multiplications, but multiplication by noninteger constants can also be essential. The purpose of this exercise is to develop an appropriate theory of constants. Let us say that vectors v_1, ..., v_s of real numbers are *Z-dependent* if there are integers $(k_1, \ldots, k_s)$ such that $\gcd(k_1, \ldots, k_s) = 1$ and $k_1v_1 + \cdots + k_sv_s$ is an all-integer vector. If no such $(k_1, \ldots, k_s)$ exist, the vectors v_1, ..., v_s are *Z-independent*.

a) Prove that if the columns of an $r \times s$ matrix V are Z-independent, so are the columns of VU, when U is any $s \times s$ unimodular matrix (a matrix of integers whose determinant is ± 1).

b) Let V be an $r \times s$ matrix with Z-independent columns. Prove that a polynomial chain to evaluate the elements of Vx from inputs x_1, ..., x_s, where $x = (x_1, \ldots, x_s)^T$, needs at least s multiplications.

c) Let V be an $r \times t$ matrix having s columns that are Z-independent. Prove that a polynomial chain to evaluate the elements of Vx from inputs $x_1, \ldots, x_t$, where $x = (x_1, \ldots, x_t)^T$, needs at least s multiplications.

d) Show how to compute the pair of values $\{x/2 + y,\ x + y/3\}$ from x and y using only one multiplication, although two multiplications are needed to compute the pair $\{x/2 + y,\ x + y/2\}$.

Page 507 lines 4 and 5 after (5) ———— 28 Apr 1995

a much simpler ... powers has been suggested ... 15.] $\rightsquigarrow$ Leonhard Euler discovered a much simpler ... powers [*Introductio in Analysin Infinitorum* **1** (1748), §76]:

Page 508 replacement for (12) and the two previous lines ———— 01 May 1995

$$W_n = \frac{1}{n}\,[t^{n-1}]\,(1 + V_2 t + V_3 t^2 + \cdots)^{-n}\,. \qquad (12)$$

Also, four lines further down: Relation (12) $\rightsquigarrow$ Relation (12), which is proved algorithmically in exercise 16,

Page 509 top ———— 29 Apr 1995

line 3: $V_{k+1} \rightsquigarrow U_0 V_{k+1}$
line 5: $nV_n \rightsquigarrow nU_0 V_n$

Page 511 near the top ———— 02 May 1995

line 12: $O(n^3) \rightsquigarrow O(N^3)$
line 14: algorithm whose running time was $\rightsquigarrow$ algorithm essentially equivalent to the method of exercise 16, with running time

Page 511 better notation for iteration ———— 03 May 1995

line 2 before (18): $f^0(x) \rightsquigarrow f^{[0]}(x)$
line 1 before (18): $f^n(x) = f\big(f^{n-1}(x)\big) \rightsquigarrow f^{[n]}(x) = f\big(f^{[n-1]}(x)\big)$
(18): $f^{m+n}(x) = f^m\big(f^n(x)\big) \rightsquigarrow f^{[m+n]}(x) = f^{[m]}\big(f^{[n]}(x)\big)$
lines 1–4 after (18): $f^n \quad f^{-n} \quad f^{-1} \quad \rightsquigarrow \quad f^{[n]} \quad f^{[-n]} \quad f^{[-1]}$ (6 changes)
lines 4 and 2 before (19): $V^{-1} \rightsquigarrow V^{[-1]}$
(20) and the next 5 lines: U^n and $U^{1/2} \rightsquigarrow U^{[n]}$ and $U^{[1/2]}$ (7 changes)

Page 511 line 5 after (18) ———— 12 Apr 1996

Eqs. (10) and (11) $\rightsquigarrow$ Eqs. (10) and (11)

▸**Page 512** line 14 ———— 14 Jul 1997

$+\,U_3 = U_1 V_4 \rightsquigarrow \ +\,U_4 = U_1 V_4$

Page 513 continuing the notational change ———— 03 May 1995

line 2 before (27) and line 1 before (28): $U^n(z) \rightsquigarrow U^{[n]}(z)$
line 4 after (28): $U^k(z) \rightsquigarrow U^{[k]}(z)$ (twice)
lines 7–8 after (28): $U^n(z) = V^{-1}\big(U_k^{(k^n-1)/(k-1)} V(z)^{k^n}\big) \rightsquigarrow$
$U^{[n]}(z) = V^{[-1]}\big(U_k^{[(k^n-1)/(k-1)]} V(z)^{k^n}\big)$
line −2: $U^n(z) = \hat{U}^n(z-\alpha) + \alpha \rightsquigarrow U^{[n]}(z) = \hat{U}^{[n]}(z-\alpha) + \alpha$

Page 513 bottom line ——— 02 Apr 1992

(1980), 54–66]. $\rightsquigarrow$ (1980), 54–66]; the V function of (27) had previously been considered by M. Kuczma, *Functional Equations in a Single Variable* (Warsaw, 1968), Lemma 9.4.

Page 514 exercise 4 ——— 29 Apr 1995

$e^{V(z)}$ and $\ln(1+V(z))$, when $V(z) = V_1z + V_2z^2 + \cdots$. $\rightsquigarrow$ $e^{V(z)}$ when $V_0 = 0$, and $\ln V(z)$ when $V_0 = 1$.

Page 514 exercise 8 ——— 01 May 1995

Lagrange's inversion ... Extend $\rightsquigarrow$ If $W(z) = W_1z + W_2z^2 + W_3z^3 + \cdots = G_1t + G_2t^2 + G_3t^3 + \cdots = G(t)$, where $z = V_1t + V_2t^2 + V_3t^3 + \cdots$ and $V_1 \neq 0$, Lagrange proved that

$$W_n = \frac{1}{n}[t^{n-1}]\, G'(t)/(V_1 + V_2t + V_3t^2 + \cdots)^n\,.$$

(Equation (12) is the special case $G_1 = V_1 = 1$, $G_2 = G_3 = \cdots = 0$.) Extend

Page 515 exercise 14 ——— 03 May 1995

$U^n(z)$ $\rightsquigarrow$ $U^{[n]}(z)$

Page 515 new exercise ——— 03 May 1995

15. [*HM20*] For what functions $U(z)$ does $V(z)$ have the simple form z^k in (27)? What do you deduce about the iterates of $U(z)$?

Page 515 new exercise ——— 01 May 1995

16. [*HM21*] Let $W(z) = G(t)$ as in exercise 8. The "obvious" way to find the coefficients W_1, W_2, W_3, ... is to proceed as follows: Set $n \leftarrow 1$ and $R_1(t) \leftarrow G(t)$. Then preserve the relation $W_nV(t) + W_{n+1}V(t)^2 + \cdots = R_n(t)$ by repeatedly setting $W_n \leftarrow [t]\, R_n(t)/V_1$, $R_{n+1}(t) \leftarrow R_n(t)/V(t) - W_n$, $n \leftarrow n+1$.

Prove Lagrange's formula of exercise 8 by showing that

$$\frac{1}{n}[t^{n-1}]\, R'_{k+1}(t)\, t^n/V(t)^n = \frac{1}{n+1}[t^n]\, R'_k(t)\, t^{n+1}/V(t)^{n+1}, \quad \text{for all } n \geq 1 \text{ and } k \geq 1.$$

Page 515 new exercises ——— 04 May 1995

▶**17.** [*M20*] Given the power series $V(z) = V_1z + V_2z^2 + V_3z^3 + \cdots$, we define the *power matrix* of V as the infinite array of coefficients $v_{nk} = \frac{n!}{k!}[z^n]V(z)^k$; the nth *poweroid* of V is then defined to be $V_n(x) = v_{n0} + v_{n1}x + \cdots + v_{nn}x^n$. Prove that poweroids satisfy the general convolution law

$$V_n(x+y) = \sum_k \binom{n}{k} V_k(x)V_{n-k}(y)\,.$$

(For example, when $V(z) = z$ we have $V_n(x) = x^n$, and this is the binomial theorem. When $V(z) = \ln(1/(1-z))$ we have $v_{nk} = \left[{n \atop k}\right]$ by Eq. 1.2.9–(26); hence the poweroid $V_n(x)$ is $x^{\overline{n}}$, and the identity is the result proved in exercise 1.2.6–33. When $V(z) = e^z - 1$ we have $V_n(x) = \sum_k \left\{{n \atop k}\right\} x^k$ and the formula is equivalent to

$$\binom{l+m}{m}\left\{{n \atop l+m}\right\} = \sum_k \binom{n}{k}\left\{{k \atop l}\right\}\left\{{n-k \atop m}\right\},$$

an identity we haven't seen before. An amazing variety of triangular arrays of coefficients that arise in combinatorial mathematics and the analysis of algorithms turn out to be the power matrices of power series.)

18. [*HM22*] Continuing exercise 17, prove that

$$xV_n(x+y) = (x+y)\sum_k \binom{n-1}{k-1} V_k(x)V_{n-k}(y)\,.$$

[*Hint:* Consider the derivative of $e^{xV(z)}$.]

19. [*M25*] Continuing exercise 17, express all the numbers v_{nk} in terms of the numbers $v_n = v_{n1} = n!\,V_n$ of the first column, and find a simple recurrence by which all columns can be computed from the sequence v_1, v_2, … . Show in particular that if all the v_n are integers, then all the v_{nk} are integers.

20. [*M20*] Continuing exercise 17, suppose we have $W(z) = U\bigl(V(z)\bigr)$ and $U_0 = 0$. Prove that the power matrix of W is the product of the power matrices of V and U: $w_{nk} = \sum_j v_{nj}u_{jk}$.

▸**21.** [*HM27*] Continuing the previous exercises, suppose $V_1 \neq 0$ and let $W(z) = -V^{[-1]}(-z)$. The purpose of this exercise is to show that the power matrices of V and W are "dual" to each other; for example, when $V(z) = \ln\bigl(1/(1-z)\bigr)$ we have $V^{[-1]}(z) = 1 - e^{-z}$, $W(z) = e^z - 1$, and the corresponding power matrices are the well-known Stirling triangles $v_{nk} = \left[{n \atop k}\right]$, $w_{nk} = \left\{{n \atop k}\right\}$.

(a) Prove that the inversion formulas for Stirling numbers, Eqs. 1.2.6–(43), hold in general:

$$\sum_k v_{nk}w_{km}(-1)^{n-k} = \sum_k w_{nk}v_{km}(-1)^{n-k} = \delta_{mn}\,.$$

(b) The relation $v_{n(n-k)} = n^{\underline{k}}[z^k]\bigl(V(z)/z\bigr)^{n-k}$ shows that, for fixed k, $v_{n(n-k)}/V_1^n$ is a polynomial in n of degree $\le 2k$. We can therefore define $v_{\alpha(\alpha-k)} = \alpha^{\underline{k}}[z^k]\bigl(V(z)/z\bigr)^{\alpha-k}$ for arbitrary α when k is a nonnegative integer, as we did for Stirling numbers in Section 1.2.6. Prove that $v_{(-k)(-n)} = w_{nk}$.

▸**22.** [*HM27*] Given $U(z) = U_0 + U_1z + U_2z^2 + \cdots$ with $U_0 \neq 0$, the αth *induced function* $U^{\{\alpha\}}(z)$ is the power series $V(z)$ defined implicitly by the equation

$$V(z) = U\bigl(zV(z)^\alpha\bigr)\,.$$

(a) Prove that $U^{\{0\}}(z) = U(z)$ and $U^{\{\alpha\}\{\beta\}}(z) = U^{\{\alpha+\beta\}}(z)$.

(b) Let $B(z)$ be the simple binomial series $1+z$. Where have we seen $B^{\{2\}}(z)$ before?

(c) Prove that $[z^n]\,U^{\{\alpha\}}(z)^x = \frac{x}{x+n\alpha}\,[z^n]\,U(z)^{x+n\alpha}$. *Hint:* If $W(z) = z/U(z)^\alpha$, we have $U^{\{\alpha\}}(z) = \bigl(W^{[-1]}(z)/z\bigr)^{1/\alpha}$.

(d) Consequently any poweroid $V_n(x)$ satisfies not only the identities of exercises 17 and 18, but also

$$\frac{(x+y)V_n(x+y+n\alpha)}{x+y+n\alpha} = \sum_k \binom{n}{k} \frac{xV_k(x+k\alpha)}{x+k\alpha}\,\frac{yV_{n-k}\bigl(y+(n-k)\alpha\bigr)}{y+(n-k)\alpha}\,;$$

$$\frac{V_n(x+y)}{y-n\alpha} = (x+y)\sum_k \binom{n-1}{k-1} \frac{V_k(x+k\alpha)}{x+k\alpha}\,\frac{V_{n-k}(y-k\alpha)}{y-k\alpha}\,.$$

[Special cases include Abel's binomial theorem, Eq. 1.2.6–(16); Rothe's identities 1.2.6–(26) and 1.2.6–(31); Torelli's sum, exercise 1.2.6–34.]

23. [*HM30*] (E. Jabotinski.) Continuing in the same vein, suppose that $U(z) = z + U_2z^2 + \cdots$ and that $U = (u_{nk})$ is the power matrix of $U(z)$. Let $u_n = u_{n1} = n!\,U_n$.

(a) Explain how to compute a matrix $\ln U$ so that the power matrix of $U^{[\alpha]}(z)$ is $\exp(\alpha \ln U) = I + \alpha \ln U + (\alpha \ln U)^2/2! + \cdots$.

(b) Let l_{nk} be the entry in row n and column k of $\ln U$, and let

$$l_n = l_{(n+1)1}, \qquad L(z) = l_1\frac{z^2}{2!} + l_2\frac{z^3}{3!} + l_3\frac{z^4}{4!} + \cdots.$$

Prove that $l_{nk} = \binom{n}{k-1} l_{n-k}$ for $1 \le k \le n$. [*Hint:* $U^{[\epsilon]}(z) = z + \epsilon L(z) + O(\epsilon^2)$.]

(c) Considering $U^{[\alpha]}(z)$ as a function of both α and z, prove that

$$\frac{\partial}{\partial\alpha}U^{[\alpha]}(z) = L(z)\frac{\partial}{\partial z}U^{[\alpha]}(z) = L\bigl(U^{[\alpha]}(z)\bigr).$$

(Consequently $L(z) = (l_{k-1}/k!)V(z)$, where $V(z)$ is the function in (27) and (28).)

(d) Show that if $u_2 \ne 0$, the numbers l_n can be computed from the recurrence

$$l_1 = u_2, \qquad \sum_{k=1}^{n-1}\binom{n}{k+1} l_k u_{n-k} = \sum_{k=1}^{n-1} l_k u_{n(k+1)}.$$

How would you use this recurrence when $u_2 = 0$?

(e) Prove the identity

$$u_n = \sum_{m=0}^{n-1}\frac{n!}{m!}\sum_{\substack{k_1+\cdots+k_m=n-1\\ k_1,\ldots,k_m\ge 1}}\frac{n_0}{(1+k_1)!}\frac{n_1}{(1+k_2)!}\cdots\frac{n_{m-1}}{(1+k_m)!}\,l_{k_1}l_{k_2}\ldots l_{k_m},$$

where $n_j = 1 + k_1 + \cdots + k_j$.

24. [*HM25*] Given the power series $U(z) = U_1z + U_2z^2 + \cdots$, where U_1 is not a root of unity, let $U = (u_{nk})$ be the power matrix of $U(z)$. (a) Explain how to compute a matrix $\ln U$ so that the power matrix of $U^{[\alpha]}(z)$ is $\exp(\alpha\ln U) = I + \alpha\ln U + (\alpha\ln U)^2/2! + \cdots$. (b) Show that if $W(z)$ is not identically zero and if $U\bigl(W(z)\bigr) = W\bigl(U(z)\bigr)$, then $W(z) = U^{[\alpha]}(z)$ for some complex number α.

25. [*M24*] If $U(z) = z + U_kz^k + U_{k+1}z^{k+1} + \cdots$ and $V(z) = z + V_lz^l + V_{l+1}z^{l+1} + \cdots$, where $k \ge 2$, $l \ge 2$, $U_k \ne 0$, $V_l \ne 0$, and $U\bigl(V(z)\bigr) = V\bigl(U(z)\bigr)$, prove that we must have $k = l$ and $V(z) = U^{[\alpha]}(z)$ for $\alpha = V_k/U_k$.

Page 515 new exercise ——— 16 May 1995

26. [*M22*] (D. Zeilberger.) Find a recurrence analogous to (9) for computing the coefficients of $W(z) = V(z)V(qz)\ldots V(q^{m-1}z)$, given q, m, and the coefficients of $V(z) = 1 + V_1z + V_2z^2 + \cdots$. Assume that q is not a root of unity.

Page 515 new exercise ——— 29 Apr 1995

▸**27.** [*HM26*] A *Dirichlet series* is a sum of the form $V(z) = V_1/1^z + V_2/2^z + V_3/3^z + \cdots$; the product $U(z)V(z)$ of two such series is the Dirichlet series $W(z)$ where

$$W_n = \sum_{d\backslash n} U_d V_{n/d}.$$

Ordinary power series are special cases of Dirichlet series, since we have $V_0 + V_1 z + V_2 z^2 + V_3 z^3 + \cdots = V_0/1^s + V_1/2^s + V_2/4^s + V_3/8^s + \cdots$ when $z = 2^{-s}$. In fact, Dirichlet series are essentially equivalent to power series $V(z_1, z_2, \ldots)$ in arbitrarily many variables, where $z_k = p_k^{-s}$ and p_k is the kth prime number.

Find recurrence relations that generalize (9) and the formulas of exercise 4, assuming that a Dirichlet series $V(z)$ is given and that we want to calculate (a) $W(z) = V(z)^\alpha$ when $V_1 = 1$; (b) $W(z) = \exp V(z)$ when $V_1 = 0$; (c) $W(z) = \ln V(z)$ when $V_1 = 1$. [*Hint:* Let $t(n)$ be the total number of prime factors of n, including multiplicity, and let $\delta \sum_n V_n/n^z = \sum_n t(n)V_n/n^z$. Show that δ is analogous to a derivative; for example, $\delta e^{V(z)} = e^{V(z)}\delta V(z)$.]

Page 516, replacement for lines 10–13 ——— 25 Mar 1995

3. (Solution by Roger Frye, after about 110 hours of computation on a Connection Machine.) $95800^4 + 217519^4 + 414560^4 = 422481^4$.

4. (One of the readers of the preliminary manuscript for this book reported that he had found a truly remarkable proof, but unfortunately the margin of his copy was too small to contain it.)

Page 517 new paragraph after line 10 ——— 14 Apr 1996

It is, however, legitimate to take ten readings $(m_0, \ldots, m_9)$ and output j if m_j is strictly less than m_i for all $i \neq j$; try again if the minimum value appears more than once. (See (h).)

▸**Page 517** line 4 of answer 2 ——— 23 Mar 1982

$2.55 \times 10^{-26} \leadsto 2.56 \times 10^{-26}$

Page 517 answer 4 ——— 14 Apr 1995

line 1: Step K11 $\leadsto$ (a) Step K11
new paragraph: (b) If X is initially 3830951656, the computation is like many of the steps in Table 1 except that we reach step K11 with $Y = 3$ instead of $Y = 5$; hence $3830951656 \to 5870802097$. Similarly, $5870802097 \to 1226919902 \to 3172562687 \to 3319967479 \to 6065038420 \to 6065038420 \to \cdots$.

▸**Page 517** answer 6 ——— 27 Dec 1983

line 6: $r \geq n \leadsto r > n$
line 10: $X_0 = a \leadsto X_1 = a$
line 12: $0 \leq i \leq \mu \leadsto 0 < i < \mu$

▸**Page 518** replacement for lines 17–21 ——— 14 Jul 1995

$\lambda = n - \max\{\, l \mid l < n \text{ and } e(l) = k\}$. After X_n has been stored in $T_{e(n)}$, it is subsequently compared with X_{n+1}, X_{n+2}, $\ldots$, $X_{n+2^{e(n)+1}}$. Therefore the procedure stops immediately after generating $X_{\mu+\lambda+j}$, where $j \geq 0$ is minimum with $e(\mu+j) \geq \lceil \lg \lambda \rceil - 1$. No X value is generated more than twice, and at most $\max(1, \lambda - 1)$ values occur more than once.

R. Sedgewick, T. G. Szymanski, and A. C. Yao have analyzed a more complex algorithm based on parameters $m \geq 2$ and $g \geq 1$: An auxiliary table of size m contains X_0, X_b, $\ldots$, X_{qb} at the moment that X_n is computed, where $b = 2^{\lceil \lg n/m \rceil}$ and $q = \lceil n/b \rceil - 1$. If $n \bmod gb < b$, X_n is compared to the entries in the table; eventually equality occurs, and we can reconstruct μ and λ after doing at most $(g+1)2^{\lceil \lg(\mu+\lambda) \rceil + 1}$ further evaluations of f. If the evaluation of f costs τ units of time, and if testing

X_n for membership in the table costs σ units, then g can be chosen so that the total running time is $(\mu+\lambda)(\tau+O(\frac{\sigma\tau}{m})^{1/2})$; this is optimum if $\sigma/\tau = O(m)$. Moreover, X_n is not computed unless $\mu+\lambda > mn/(m+4g+2)$, so we can use this method "online" to output elements that are guaranteed to be distinct, making only $2+O(m^{-1/2})$ function evaluations per output. [*SICOMP* **11** (1982), 376–390.]

▸**Page 519** line 2 ——— 05 May 1981

$6\sqrt{\pi/2m} \leadsto \sqrt{\pi/2m}$

Page 519 last line of answer 13 ——— 15 Mar 1995

The average value of $\max(\mu+\lambda)$ is still unknown.) $\leadsto$ The average value of $\max\mu$ is asymptotic to $Q(m)\ln 4$, and the average value of $\max(\mu+\lambda)$ is asymptotic to $1.9268Q(m)$; see Flajolet and Odlyzko, *Lecture Notes in Comp. Sci.* **434** (1990), 329–354.)

Page 519 replacement for lines 6–9 of answer 14 ——— 30 Jul 1984

exercise 2.3.4.4–17. The value of $c_r(m)$ may be determined as in exercise 13:

$$c_r(m) = \sum_{n\ge 1} T_{mn} \left[{n \atop r}\right] = m^{m-1}\left(\frac{1}{0!}\left[{1 \atop r}\right] + \frac{1}{1!}\left[{2 \atop r}\right]\frac{m-1}{m} + \frac{1}{2!}\left[{3 \atop r}\right]\frac{m-1}{m}\frac{m-2}{m} + \cdots\right).$$

▸**Page 520** answer 15 ——— 01 Feb 1981

$(m-1)^m/m \leadsto (m-1)^m/m^m$

Page 520 new answers ——— 14 Apr 1995

20. It suffices to consider the simpler mapping $g(X)$ defined by steps K2–K13. Working backward from 6065038420, we obtain a total of 597 solutions; the smallest is 0009612809 and the largest is 9995371004.

21. We may work with $g(X)$ as in the previous exercise, but now we want to run the function forward instead of backward. There is an interesting tradeoff between time and space. Using a few megabytes of memory, the author verified the conjecture for all starting values less than 0000165181 in 1994, then decided to wait until computers get bigger and faster. Notice that the mechanism of step K1 tends to make the period length small. So does the existence of X's with large in-degree; for example, 512 choices of $X = {*}6{*}{*}{*}{*}{*}{*}{*}{*}$ in step K2 will go to K10 with $X \leftarrow 0500000000$. Algorithm K is a *terrible* random number generator.

Page 520 new answer ——— 13 Jul 1995

22. If f were truly random, this would be ideal; but how do we construct such f? The function defined by Algorithm K would work much better under this scheme, although it does have decidedly nonrandom properties (see the previous answer).

Page 520 new answer ——— 26 Apr 1995

23. The function f permutes its cyclic elements; let $(x_0,\ldots,x_{k-1})$ be the "unusual" representation of the inverse of that permutation. Then proceed to define $x_k,\ldots,x_{m-1}$ as in exercise 2.3.4.4–18. [Cf. *J. Combinatorial Theory* **8** (1970), 361–375.]

For example, if $m = 10$ and $\bigl(f(0),\ldots,f(9)\bigr) = (3,1,4,1,5,9,2,6,5,4)$, we have $(x_0,\ldots,x_9) = (4,9,5,1,1,3,4,2,6,5)$; if $(x_0,\ldots,x_9) = (3,1,4,1,5,9,2,6,5,4)$, we have $\bigl(f(0),\ldots,f(9)\bigr) = (6,4,9,3,1,1,2,5,4,5)$.

Page 521 replacement for answer 3 ——— 27 Aug 1995

3. Let $a' = aw \bmod m$, and let m' be such that $mm' \equiv 1$ (modulo w). Set $y \leftarrow \mathrm{lomult}(a', x)$, $z \leftarrow \mathrm{himult}(a', x)$, $t \leftarrow \mathrm{lomult}(m', y)$, $u \leftarrow \mathrm{himult}(m, t)$. Then we have $mt \equiv a'x$ (modulo w), hence $a'x - mt = (z - u)w$, hence $ax \equiv z - u$ (modulo m); it follows that $ax \bmod m = z - u + [z < u]m$.

Page 522 replacement for answer 9 ——— 30 Aug 1995

9. (a) Both sides equal $aq\lfloor x/q \rfloor$.

(b) Set $t \leftarrow a(x \bmod q) - r\lfloor x/q \rfloor$, where $r = m \bmod a$; the constants q and r can be precomputed. Then $ax \bmod m = t + [t < 0]m$, because we can prove that $t \ge -m$: Clearly $a(x \bmod q) \le a(q-1) < m$. Also $r\lfloor x/q \rfloor \le r\lfloor (m-1)/q \rfloor = r\lfloor a + (r-1)/q \rfloor = ra \le qa < m$ if $0 < r \le q$; and $a^2 \le m$ implies $r < a \le q$. [This technique is implicit in a program published by B. A. Wichmann and I. D. Hill, *Applied Stat.* **31** (1982), 190.]

(c) If $r > q$ and $x = m - 1$ we have $r\lfloor x/q \rfloor \ge (q+1)(a+1) > m$. So the condition $r \le q$ is necessary and sufficient for validity of method (b); this means $\frac{m}{q} - 1 \le a \le \frac{m}{q}$. Let $t = \lfloor \sqrt{m} \rfloor$. The intervals $[\frac{m}{q} - 1 \,..\, \frac{m}{q}]$ are disjoint for $1 \le q \le t$, and they include exactly 1 or 2 integers, depending on whether q is a divisor of m. These intervals account for all solutions with $a > \sqrt{m}$; they also include the case $a = t$, if $(\sqrt{m} \bmod 1) < \frac{1}{2}$, and the case $a = t - 1$ if $m = t^2$. Thus the total number of "lucky" multipliers is exactly $2\lfloor \sqrt{m} \rfloor + \lfloor d(m)/2 \rfloor - [(\sqrt{m} \bmod 1) < \frac{1}{2}] - 1$, where $d(m)$ is the number of divisors of m.

10. We can assume that $a \le \frac{1}{2}m$; otherwise we can obtain $ax \bmod m$ from $(m-a)x \bmod m$. Then we can represent $a = a'a'' - a'''$, where a', a'', and a''' are all less than $\sqrt{m}$; for example, we can take $a' = \lfloor \sqrt{m} \rfloor$ and $a'' = \lceil a/a' \rceil$. It follows that $ax \bmod m$ is $\bigl(a'(a''x \bmod m) \bmod m - (a'''x \bmod m)\bigr) \bmod m$, and the inner three operations can all be handled by exercise 9.

When $m = 2^{31} - 1$ we can take advantage of the fact that $m - 1$ has 192 divisors to find cases in which $m = q'a' + 1$, simplifying the general method because $r' = 1$. It turns out that 86 of these divisors lead to lucky a'' and a''', when $a = 62089911$; the best such case is probably $a' = 3641$, $a'' = 17053$, $a''' = 62$, because 3641 and 62 both divide $m - 1$. This decomposition yields the scheme

$$\begin{aligned} t &\leftarrow 17053(x \bmod 125929) - 16410\lfloor x/125929 \rfloor\,,\\ t &\leftarrow 3641(t \bmod 589806) - \lfloor t/589806 \rfloor\,,\\ t &\leftarrow t - \bigl(62(x \bmod 34636833) - \lfloor x/34636833 \rfloor\bigr)\,, \end{aligned}$$

where '$-$' denotes subtraction mod m. The 'mod' operations count as one multiplication and one subtraction, because $x \bmod q = x - q\lfloor x/q \rfloor$ and the operation $\lfloor x/q \rfloor$ has already been done; thus, we have performed seven multiplications, three divisions, and seven subtractions. But it's even better to notice that 62089911 itself has 24 divisors; they lead to five suitable factorizations with $a''' = 0$. For example, when $a' = 883$ and $a'' = 70317$ we need only six multiplications, two divisions, four subtractions:

$$\begin{aligned} t &\leftarrow 883(x \bmod 2432031) - 274\lfloor x/2432031 \rfloor\,,\\ t &\leftarrow 70317(t \bmod 30540) - 2467\lfloor t/30540 \rfloor\,. \end{aligned}$$

[Can the worst-case number of multiplications plus divisions be reduced to at most 11, for all a and m, or is 12 the best upper bound? Another way to achieve 12 appears in exercise 4.3.3–17.]

Page 522 new answers ________ 16 Sep 1995

11. (a) Let $m = 9999998999 = 10^{10} - 10^3 - 1$. To multiply $(x_9x_8 \ldots x_0)_{10}$ by 10 modulo m, use the fact that $10^{10}x_9 \equiv 10^3 x_9 + x_9$: Add $(x_9000)_{10}$ to $(x_8x_7 \ldots x_0x_9)_{10}$. And to avoid circular shifting, imagine that the digits are arranged on a wheel: Just add the high-order digit x_9 to the digit x_2 three positions left, and point to x_8 as the new high-order digit. If $x_9 + x_2 \geq 10$, a carry propagates to the left. And if this carry ripples all the way to the left of x_8, it propagates not only to x_9 but also to the x_2 position; it may continue to propagate from both x_9 and x_2 before finally settling down. (The numbers might also become slightly larger than m. For example, 0999999900 goes to $9999999000 = m + 1$, which goes to $9999999009 = m + 10$. But a redundant representation isn't necessarily harmful.)

(b) This is the operation of *dividing* by 10, so we do the opposite of (a): Move the high-order digit pointer cyclically *left*, and *subtract* the new high-order digit from the digit three to its left. If the result of subtraction is negative, "borrow" in the usual fashion (Algorithm 4.3.1S); i.e., decrease the preceding digit by 1. Borrowing may propagate as in (a), but never past the high-order digit position. This operation keeps the numbers nonnegative and less than m. (Thus, division by 10 turns out to be easier than multiplication by 10.)

(c) We can *remember* the borrow-bit instead of propagating it, because it can be incorporated into the subtraction on the next step. Thus, if we define digits x_n and borrow-bits b_n by the recurrence

$$x_n = (x_{n-10} - x_{n-3} - b_n) \bmod 10 = x_{n-10} - x_{n-3} - b_n + 10b_{n+1},$$

we have $9999999000^n \bmod 9999998999 = X_n$ by induction on n, where

$$\begin{aligned} X_n &= (x_{n-1}x_{n-2}x_{n-3}x_{n-4}x_{n-5}x_{n-6}x_{n-7}x_{n+2}x_{n+1}x_n)_{10} - 1000b_{n+3} \\ &= (x_{n-1}x_{n-2} \ldots x_{n-10})_{10} - (x_{n-1}x_{n-2}x_{n-3})_{10} - b_n, \end{aligned}$$

provided that the initial conditions are set up to make $X_0 = 1$. Notice that $10X_{n+1} = (x_nx_{n-1}x_{n-2}x_{n-3}x_{n-4}x_{n-5}x_{n-6}x_{n+3}x_{n+2}x_{n+1}0)_{10} - 10000b_{n+4} = mx_n + X_n$; it follows that $0 \leq X_n < m$ for all $n \geq 0$.

(d) If $0 \leq U < m$, the first digit of the decimal representation of U/m is $\lfloor 10U/m \rfloor$, and the subsequent digits are the decimal representation of $(10U \bmod m)/m$; see, for example, Method (2a) in Section 4.4. Thus $U/m = (.u_1u_2\ldots)_{10}$ if we set $U_0 = U$ and $U_n = 10U_{n-1} \bmod m = 10U_{n-1} - mu_n$. Informally, the digits of $1/m$ are the leading digits of $10^n \bmod m$ for $n = 1, 2, \ldots$, a sequence that is eventually periodic; these are the leading digits of $10^{-n} \bmod m$ in reverse order, so we have calculated them in (c).

A rigorous proof is, of course, preferable to handwaving. Let λ be the least positive integer with $10^\lambda \equiv 1$ (modulo m), and define $x_n = x_{n \bmod \lambda}$, $b_n = b_{n \bmod \lambda}$, $X_n = X_{n \bmod \lambda}$ for all $n < 0$. Then the recurrences for x_n, b_n, and X_n in (c) are valid for all integers n. If $U_0 = 1$ it follows that $U_n = X_{-n}$ and $u_n = x_{-n}$; hence

$$\frac{999999900^n \bmod 9999998999}{9999998999} = (.x_{n-1}x_{n-2}x_{n-3}\ldots)_{10}.$$

(e) Let w be the computer's word size w, and use the recurrence

$$x_n = (x_{n-k} - x_{n-l} - b_n) \bmod w = x_{n-k} - x_{n-l} - b_n + wb_{n+1},$$

where $0 < l < k$ and k is large. Then $(.x_{n-1}x_{n-2}x_{n-3}\ldots)_w = X_n/m$, where $m = w^k - w^l - 1$ and $X_{n+1} = (w^{k-1} - w^{l-1})X_n \bmod m$. The relation

$$X_n = (x_{n-1}\ldots x_{n-k})_w - (x_{n-1}\ldots x_{n-l})_w + b_n$$

holds for $n \geq 0$; the values of $x_{-1}, \ldots, x_{-k}$, and b_0 should be such that $0 \leq X_0 < m$.

Such random number generators, and the similar ones in the following exercise, were introduced by G. Marsaglia and A. Zaman [*Annals of Applied Probability* **1** (1991), 462–480], who called the method *subtract-with-borrow*. Their starting point was the radix-w representation of fractions with denominator m. The relation to linear congruential sequences was noticed by Shu Tezuka, and analyzed in detail by Tezuka, L'Ecuyer, and Couture [*ACM Trans. Modeling and Computer Simulation* **3** (1993), 315–331]. The period length is discussed in exercise 3.2.1.2–22.

12. Multiplication by 10 now requires *negating* the digit that is added. For this purpose it is convenient to represent a number with its last three digits negated; for example, $9876543210 = (9876544\bar{7}\bar{9}\bar{0})_{10}$. Then 10 times $(x_9 \ldots x_3\bar{x}_2\bar{x}_1\bar{x}_0)_{10}$ is $(x_8 \ldots x_3x'\bar{x}_1\bar{x}_0\bar{x}_9)_{10}$ where $x' = x_9 - x_2$. Similarly, $(x_9 \ldots x_3\bar{x}_2\bar{x}_1\bar{x}_0)_{10}$ divided by 10 is $(x_0x_9 \ldots x_4\bar{x}''\bar{x}_2\bar{x}_1)_{10}$ where $x'' = x_0 - x_3$. The recurrence

$$x_n = (x_{n-3} - x_{n-10} - b_{n-1}) \bmod 10 = x_{n-3} - x_{n-10} - b_{n-1} + 10b_n$$

yields $8999999101^n \bmod 9999999001 = X_n$ where

$$\begin{aligned} X_n &= (x_{n-1}x_{n-2}x_{n-3}x_{n-4}x_{n-5}x_{n-6}x_{n-7}\bar{x}_{n+2}\bar{x}_{n-1}\bar{x}_n)_{10} + 1000b_{n+3} \\ &= (x_{n-1}x_{n-2}\ldots x_{n-10})_{10} - (x_{n-1}x_{n-2}x_{n-3})_{10} + b_n. \end{aligned}$$

When the radix is generalized from 10 to w, we find that the inverse powers of w modulo $w^k - w^l + 1$ are generated by

$$x_n = (x_{n-l} - x_{n-k} - b_n) \bmod w = x_{n-l} - x_{n-k} - b_n + wb_{n+1}$$

(the same as in exercise 11 but with k and l interchanged).

13. Mild generalization: We can effectively divide by b modulo $b^k - b^l \pm 1$ for any b less than or equal to the word size w, since the recurrence for x_n is almost as efficient when $b < w$ as it is when $b = w$. Strong generalization: The recurrence

$$x_n = (a_1x_{n-1} + \cdots + a_kx_{n-k} + c_n) \bmod b, \qquad c_{n+1} = \left\lfloor \frac{a_1x_{n-1} + \cdots + a_kx_{n-k} + c_n}{b} \right\rfloor$$

is equivalent to $X_n = b^{-1}X_{n-1} \bmod |m|$ in the sense that $X_n/|m| = (.x_{n-1}x_{n-2}\ldots)_b$, if we define

$$m = a_kb^k + \cdots + a_1b - 1 \qquad \text{and} \qquad X_n = \left(\sum_{j=1}^{k} a_j(x_{n-1}\ldots x_{n-k})_b + c_n\right)(\operatorname{sign} m)\,.$$

The initial values $x_{-1}\ldots x_{-k}$ and c_0 should be selected so that $0 \leq X_0 < |m|$; we will then have $x_n = (bX_{n+1} - X_n)/|m|$ for $n \geq 0$. The values of x_j for $j < 0$ that appear in the formula $X_n/|m| = (.x_{n-1}x_{n-2}\ldots)_b$ are properly regarded as $x_{j \bmod \lambda}$, where $b^\lambda \equiv 1$ (modulo m); these values may differ from the numbers $x_{-1}, \ldots, x_{-k}$ that were initially supplied. The carry digits c_n will satisfy

$$\sum_{j=1}^{k} \min(0, a_j) \leq c_n < \sum_{j=1}^{k} \max(0, a_j)$$

if the initial carry c_0 is in this range.

The special case $m = b^k + b^l - 1$, for which $a_j = \delta_{jl} + \delta_{jk}$, is of particular interest because it can be computed so easily; Marsaglia and Zaman called this the *add-with-carry* generator:

$$x_n = (x_{n-l} + x_{n-k} + c_n) \bmod b = x_{n-l} + x_{n-k} + c_n - b\,c_{n+1}.$$

Another potentially attractive possibility is to use $k = 2$ in a generator with, say, $b = 2^{31}$ and $m = 65430b^2 + b - 1$. This modulus m is prime, and the period length turns out to be $(m-1)/2$. The spectral test of Section 3.3.4 indicates that the spacing between planes is good (large ν values), although of course the multiplier b^{-1} is poor in comparison with other multipliers for this particular modulus m.

Exercise 3.2.1.2–22 contains additional information about subtract-with-borrow and add-with-carry moduli that lead to extremely long periods.

Page 523 lines 8 and 9 ———— 02 Feb 1981

so ... result. $\leadsto$ so $(a^{2^{e-1}} - 1)/(a - 1) \equiv 0$ (modulo 2^e) iff $(a^{2^{e-1}} - 1)/2 \equiv 0$ (modulo $2^{e+1}/2$), which is true.

▸**Page 523** line 4 of answer 11 ———— 02 Feb 1981

we find that ... $\equiv 1$ $\leadsto$
we find that $(\pm x)^{2^{e-f-1}} \not\equiv 1$ (modulo 2^e), while $x^{2^{e-f}} = (\pm x)^{2^{e-f}} \equiv 1$

▸**Page 524** replacement for answer 19 ———— 29 Aug 1995

19. Several examples appear in lines 17–20 of Table 3.3.4–1.

Page 524 new answer ———— 19 Sep 1995

22. Algorithm 4.5.4P is able to test numbers of the form $m = b^k \pm b^l \pm 1$ for primality in a reasonable time when, say, $b \approx 2^{32}$ and $l < k \approx 100$; the calculations should be done in radix b so that the special form of m speeds up the operation of squaring mod m. (Consider, for example, squaring mod 9999998999 in decimal notation.) Algorithm 4.5.4P should, of course, be used only when m is known to have no small divisors.

Marsaglia and Zaman [*Annals of Applied Probability* **1** (1991), 474–475] showed that $m = b^{43} - b^{22} + 1$ is prime with primitive root b when b is the prime number $2^{32} - 5$. This required factoring $m - 1 = b^{22}(b-1)(b^6+b^5+b^4+b^3+b^2+b+1)(b^{14}+b^7+1)$ in order to establish the primitivity of b; one of the 17 prime factors of $m - 1$ has 99 decimal digits. As a result, we can be sure that the sequence $x_n = (x_{n-22} - x_{n-43} - c_n) \bmod b = x_{n-22} - x_{n-43} - c_n + bc_{n+1}$ has period length $m - 1 \approx 10^{414}$ for every nonzero choice of seed values $0 \le x_{-1}, \ldots, x_{-43} < b$ when $c_0 = 0$.

However, 43 is still a rather small value for k from the standpoint of the birthday spacings test (see Section 3.3.2J), and 22 is rather near 43/2. Considerations of "mixing" indicate that we prefer values of k and l for which the first few partial quotients in the continued fraction of l/k are small. To avoid potential problems with this generator, it's a good idea to discard some of the numbers, as recommended by Lüscher (see Section 3.2.2).

Here are some prime numbers of the form $b^k \pm b^l \pm 1$ that satisfy the mixing constraint when $b = 2^{32}$ and $50 < k \le 100$: For subtract-with-borrow, $b^{57} - b^{17} - 1$, $b^{73} - b^{17} - 1$, $b^{86} - b^{62} - 1$, $b^{88} - b^{52} - 1$, $b^{95} - b^{61} - 1$; $b^{58} - b^{33} + 1$, $b^{62} - b^{17} + 1$, $b^{69} - b^{24} + 1$, $b^{70} - b^{57} + 1$, $b^{87} - b^{24} + 1$. For add-with-carry, $b^{56} + b^{22} - 1$, $b^{61} + b^{44} - 1$, $b^{74} + b^{27} - 1$, $b^{90} + b^{65} - 1$. (Less desirable from a mixing standpoint are the primes $b^{56} - b^5 - 1$, $b^{56} - b^{32} - 1$, $b^{66} - b^{57} - 1$, $b^{76} - b^{15} - 1$, $b^{84} - b^{26} - 1$, $b^{90} - b^{42} - 1$,

$b^{93}-b^{18}-1$; $b^{52}-b^{8}+1$, $b^{60}-b^{12}+1$, $b^{67}-b^{8}+1$, $b^{67}-b^{63}+1$, $b^{83}-b^{14}+1$; $b^{65}+b^{2}-1$, $b^{76}+b^{11}-1$, $b^{88}+b^{30}-1$, $b^{92}+b^{48}-1$.)

To calculate the period of the resulting sequences, we need to know the factors of $m-1$; but this is unfeasible for such large numbers unless we are extremely lucky. Suppose we do succeed in find the prime factors $q_1, \ldots, q_t$; then the probability that $b^{(m-1)/q} \bmod m = 1$ is extremely small, only $1/q$, except for the very small primes q. Therefore we can be quite confident that the period of $b^n \bmod m$ is extremely long even though we cannot factor $m-1$.

Indeed, the period is almost certainly very long even if m is not prime. Consider, for example, the case $k=10$, $l=3$, $b=10$ (which is much too small for random number generation but small enough that we can easily compute the exact results). In this case $\langle 10^n \bmod m\rangle$ has period length $\mathrm{lcm}(219, 11389520) = 2494304880$ when $m = 9999998999 = 439 \cdot 22779041$; 4999999500 when $m = 9999999001$; 5000000499 when $m = 10000000999$; and $\mathrm{lcm}(1, 16, 2686, 12162) = 130668528$ when $m = 10000001001 = 3 \cdot 17 \cdot 2687 \cdot 72973$. Rare choices of the seed values may shorten the period when m is not prime. But we can hardly go wrong if we choose, say, $k = 1000$, $l = 619$, and $b = 2^{16}$.

Page 525 line 7 ——— 19 Sep 1995

$2^{23}+2^{14}+2^2+1 \rightsquigarrow 2^{23}+2^{13}+2^2+1$

Page 526 replacement for lines −3 and −2 ——— 15 Jul 1995

$X_{n+2} \equiv X_n + 2$ (modulo 4), by trial; and if $p=3$, we have $X_{n+3} \equiv X_n - d + 3c$ (modulo 9), using (i) and (ii). For $p \ge 5$, we

Page 527 last line of answer 8 ——— 02 Aug 1995

(1967) $\rightsquigarrow$ (Philadelphia: SIAM, 1969)

Page 531 new answer ——— 16 Apr 1995

29. The following algorithm performs about k^2 operations in the worst case, but its average running time is much faster, perhaps $O(\log k)$ or even $O(1)$:

X1. Set $(a_0, a_1, \ldots, a_k) \leftarrow (x_1, \ldots, x_k, m-1)$.

X2. Let i be minimum with $a_i > 0$ and $i > 0$. Do subroutine Y for $j = i+1$, $\ldots$, k, while $a_k > 0$.

X3. If $a_0 > a_k$, $f(x_1, \ldots, x_k) = a_0$; otherwise if $a_0 > 0$, $f(x_1, \ldots, x_k) = a_0 - 1$; otherwise $f(x_1, \ldots, x_k) = a_k$. ▌

Y1. Set $l \leftarrow 0$. (The subroutine in steps Y1–Y3 essentially tests the lexicographic relation $(a_i, \ldots, a_{i+k-1}) \ge (a_j, \ldots, a_{j+k-1})$, decreasing a_k if necessary to make this inequality true. We assume that $a_{k+1} = a_1$, $a_{k+2} = a_2$, etc.)

Y2. If $a_{i+l} > a_{j+l}$, exit the subroutine. Otherwise if $j + l = k$, set $a_k \leftarrow a_{i+l}$. Otherwise if $a_{i+l} = a_{j+l}$, go on to step Y3. Otherwise if $j + l > k$, decrease a_k by 1 and exit. Otherwise set $a_k \leftarrow 0$ and exit.

Y3. Increase l by 1, and return to step Y2 if $l < k$. ▌

This problem was first solved by H. Fredricksen when $m = 2$ [*J. Combinatorial Theory* **9** (1970), 1–5; **A12** (1972), 153–154]; in that special case the algorithm is simpler and it can be done with k-bit registers. See also S. Xie, *Discrete Applied Math.* **16** (1987), 157–177.

Page 531 new answer ——— 15 Jul 1995

30. By exercise 11, it suffices to show that the period length mod 8 is $4(2^k-1)$; this will be true iff $x^{2(2^k-1)} \not\equiv 1$ (modulo 8 and $f(x)$) iff $x^{2^k-1} \not\equiv 1$ (modulo 4 and $f(x)$). Write $f(x) = f_e(x^2) + xf_o(x^2)$, where $f_e(x^2) = \frac{1}{2}(f(x) + f(-x))$. Then $f(x)^2 + f(-x)^2 \equiv 2f(x^2)$ (modulo 8) iff $f_e(x)^2 + xf_o(x)^2 \equiv f(x)$ (modulo 4); and the latter condition holds iff $f_e(x)^2 \equiv -xf_o(x)^2$ (modulo 4 and $f(x)$), because $f_e(x)^2 + xf_o(x)^2 = f(x) + O(x^{k-1})$. Furthermore, working modulo 2 and $f(x)$, we have $f_e(x)^2 \equiv f_e(x^2) \equiv xf_o(x^2) \equiv x^{2^k} f_o(x)^2$, hence $f_e(x) \equiv x^{2^{k-1}} f_o(x)$. Therefore $f_e(x)^2 \equiv x^{2^k} f_o(x)^2$ (modulo 4 and $f(x)$), and the hint follows. A similar argument proves that $x^{2^k} \equiv x$ (modulo 4 and $f(x)$) iff $f(x)^2 + f(-x)^2 \equiv 2(-1)^k f(-x^2)$ (modulo 8).

(b) The condition can hold only when l is odd and $k = 2l$. But then $f(x)$ is primitive modulo 2 only when $k = 2$. [*Math. Comp.* **63** (1994), 389–401.]

Page 531 new answer ——— 18 Aug 1995

31. We have $X_n \equiv (-1)^{Y_n} 3^{Z_n} \bmod 2^e$ for some Y_n and Z_n, by Theorem 3.2.1.2C; hence $Y_n = (Y_{n-24} + Y_{n-55}) \bmod 2$ and $Z_n = (Z_{n-24} + Z_{n-55}) \bmod 2^{e-2}$. Since Z_k is odd iff $X_k \bmod 8 = 3$ or 5, the period length is $2^{e-3}(2^{55}-1)$ by the previous exercise.

Page 531 new answers ——— 26 Sep 1995

32. We can ignore the 'mod m' and put it back afterwards. The generating function $g(z) = \sum_n X_n z^n$ is a polynomial multiple of $1/(1 - z^{24} - z^{55})$; hence $\sum_n X_{2n} z^{2n} = \frac{1}{2}(g(z) + g(-z))$ is a polynomial divided by $(1 - z^{24} - z^{55})(1 - z^{24} + z^{55}) = 1 - 2z^{24} + z^{48} - z^{110}$. The first desired recurrence is therefore $X_{2n} = (2X_{2(n-12)} - X_{2(n-24)} + X_{2(n-55)}) \bmod m$. Similarly, $\sum_n X_{3n} z^{3n} = \frac{1}{3}(g(z) + g(\omega z) + g(\omega^2 z))$ where $\omega = e^{2\pi i/3}$, and we find $X_{3n} = (3X_{3(n-8)} - 3X_{3(n-16)} + X_{3(n-24)} + X_{3(n-55)}) \bmod m$.

33. (a) $g_{n+t}(z) \equiv z^t g_n(z)$ (modulo m and $1 + z^{31} - z^{55}$), by induction on t. (b) Since $z^{500} \bmod (1 + z^{31} - z^{55}) = 792z^2 + z^5 + 17z^6 + 715z^9 + 36z^{12} + z^{13} + 364z^{16} + 210z^{19} + 105z^{23} + 462z^{26} + 16z^{30} + 1287z^{33} + 9z^{36} + 18z^{37} + 1001z^{40} + 120z^{43} + z^{44} + 455z^{47} + 462z^{50} + 120z^{54}$ (see Algorithm 4.6.1D), we have $X_{500} = (792X_2 + X_5 + \cdots + 120X_{54}) \bmod m$.

[It is interesting to compare the similar formula $X_{165} = (X_0 + 3X_7 + X_{14} + 3X_{31} + 4X_{38} + X_{45}) \bmod m$ to the sparser recurrence for $\langle X_{3n} \rangle$ in the previous exercise. Lüscher's method of generating 165 numbers and using only the first 55 is clearly superior to the idea of generating 165 and using only X_3, X_6, …, X_{165}.]

Page 531 new answers ——— 18 Sep 1995

34. Let $q_0 = 0$, $q_1 = 1$, $q_{n+1} = cq_n + aq_{n-1}$. Then we have $\begin{pmatrix} 0 & 1 \\ a & c \end{pmatrix}^n = \begin{pmatrix} aq_{n-1} & q_n \\ aq_n & q_{n+1} \end{pmatrix}$, $X_n = (q_{n+1}X_0 + aq_n)/(q_n X_0 + aq_{n-1})$, and $x^n \bmod f(x) \equiv q_n x + aq_{n-1}$, for $n \ge 1$. Thus if $X_0 = 0$ we have $X_n = 0$ iff $x^n \bmod f(x)$ is a nonzero constant.

35. Conditions (i) and (ii) imply that $f(x)$ is irreducible. For if $f(x) = (x - r_1)(x - r_2)$ and $r_1 r_2 \ne 0$ we have $x^{p-1} \equiv 1$ if $r_1 \ne r_2$ and $x^p \equiv r_1$ if $r_1 = r_2$.

Let ξ be a primitive root of a field with p^2 elements, and suppose $\xi^{2k} = c_k \xi^k + a_k$. The quadratic polynomials we seek are precisely the polynomials $f_k(x) = x^2 - c_k x - a_k$ where $1 \le k < p^2 - 1$ and $k \perp (p+1)$. (See exercise 4.6.2–16.) Each polynomial occurs for two values of k; hence the number of solutions is $\frac{1}{2}(p^2 - 1) \prod_{q \backslash p+1} (1 - 1/q)$.

36. In this case X_n is always odd, so X_n^{-1} exists mod 2^e. The sequence $\langle q_n \rangle$ defined in answer 34 is 0, 1, 2, 1, 0, 1, 2, 1, … modulo 4. We also have $q_{2n} = q_n(q_{n+1} + aq_{n-1})$ and $q_{2n-1} = aq_{n-1}^2 + q_n^2$; hence $q_{2n+1} - aq_{2n-1} = (q_{n+1} - aq_{n-1})(q_{n+1} + aq_{n+1})$. Since

$q_{n+1} + aq_{n+1} \equiv 2$ (modulo 4) when n is even, we deduce that q_{2^e} is an odd multiple of 2^e and $q_{2^e+1} - aq_{2^e-1}$ is an odd multiple of 2^{e+1}, for all $e \geq 0$. Therefore

$$q_{2^e} + aq_{2^e-1} \equiv q_{2^e+1} + aq_{2^e} + 2^{e+1} \pmod{2^{e+2}}.$$

And $X_{2^e-2} \equiv (q_{2^{e-2}+1} + aq_{2^{e-2}})/(q_{2^{e-2}} + aq_{2^{e-2}-1}) \not\equiv 1$ (modulo 2^e), while $X_{2^e-1} \equiv 1$. Conversely, we need $a \bmod 4 = 1$ and $c \bmod 4 = 2$; otherwise $X_{2n} \equiv 1$ (modulo 8). [Eichenauer, Lehn, and Topuzoğlu, *Math. Comp.* **51** (1988), 757–759.] The low-order bits of this sequence have a short period, so inversive generators with prime modulus are preferable.

37. We can assume that $b_1 = 0$. By exercise 34, a typical vector in V is

$$(x, (s'_2x + as_2)/(s_2x + as''_2), \ldots, (s'_dx + as_d)/(s_dx + as''_d)),$$

where $s_j = q_{b_j}$, $s'_j = q_{b_j+1}$, $s''_j = q_{b_j-1}$. This vector belongs to the hyperplane H iff

$$r_1x + \frac{r_2t_2}{x-u_2} + \cdots + \frac{r_dt_d}{x-u_d} \equiv r_0 - r_2s'_2s_2^{-1} - \cdots - r_ds'_ds_d^{-1} \pmod{p},$$

where $t_j = a - as'_js''_js_j^{-2} = -(-a)^{b_j}s_j^{-2}$ and $u_j = as''_js_j^{-1}$. But this relation is equivalent to a polynomial congruence of degree $\leq d$; so it cannot hold for $d+1$ values of x unless it holds for all x, including the distinct points $x = u_2, \ldots, x = u_d$. Hence $r_2 = \cdots = r_d \equiv 0$, and $r_1 \equiv 0$. [Cf. J. Eichenauer-Herrmann, *Math. Comp.* **56** (1991), 297–301.] *Notes:* If we consider the $(p+1-d) \times (d+1)$ matrix M with rows $\{(1, v_1, \ldots, v_d) \mid (v_1, \ldots, v_d) \in V\}$, this exercise is equivalent to the assertion that any $d+1$ rows of M are linearly independent modulo p. It is interesting to plot the points (X_n, X_{n+1}) for $p \approx 1000$ and $0 \leq n \leq p$; traces of circles, rather than straight lines, meet the eye.

Page 531 line −2 ——— 02 Feb 1981

$F_n(x) - F_n(y) \rightsquigarrow F_n(y) - F_n(x)$

▸**Page 532** line 6 ——— 22 Feb 1996

((Cf. $\rightsquigarrow$ (Cf.

Page 534 new answer ——— 13 Aug 1996

20. For example, the term $O(1/n)$ is $-(\frac{4}{9}s^4 - \frac{2}{3}s^2)/n + O(n^{-3/2})$. A complete expansion has been obtained by H. A. Lauwerier, *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete* **2** (1963), 61–68.

Page 534 new answer ——— 20 Aug 1995

25. (a) Since $c_{ij} = \mathrm{E}\bigl(\sum_{k=1}^n a_{ik}X_k \sum_{l=1}^n a_{jl}X_l\bigr) = \sum_{k=1}^n a_{ik}a_{jk}$, we have $C = AA^T$. (b) Consider the singular value decomposition $A = U^TDV$, where U and V are orthogonal of sizes $m \times m$ and $n \times n$, and D is $m \times n$ with entries $d_{ij} = [i=j]\sigma_j$; the singular values σ_j are all positive. [See, for example, Golub and Van Loan, *Matrix Computations* (1983), §2.3.] If $C\overline{C}C = C$ we have $SBS = S$, where $S = DD^T$ and $B = U\overline{C}U^T$. Thus $s_{ij} = [i=j]\sigma_j^2$, where we let $\sigma_{n+1} = \cdots = \sigma_m = 0$, and $s_{ij} = \sum_{k,l} s_{ik}b_{kl}s_{lj} = \sigma_i^2\sigma_j^2b_{ij}$. Consequently $b_{ij} = [i=j]/\sigma_j^2$ if $i, j \leq n$, and we deduce that D^TBD is the $n \times n$ identity matrix. Let $Y = (Y_1 - \mu_1, \ldots, Y_m - \mu_m)^T$ and $X = (X_1, \ldots, X_n)^T$; it follows that $V = Y^T\overline{C}Y = X^TA^T\overline{C}AX = X^TV^TD^TBDVX = X^TX$.

▸**Page 536** answer 15 ——— 02 Feb 1981

and S has $\rightsquigarrow$ and X has

▸**Page 536** line −5 ——— 02 Feb 1981

$$\begin{pmatrix} U_1' \, U_2' \ldots U_{n-1}' \\ V_1' \, V_2' \ldots V_{n-1}' \end{pmatrix} \leadsto \begin{pmatrix} U_0' & U_1' & \ldots & U_{n-1}' \\ V_0' & V_1' & \ldots & V_{n-1}' \end{pmatrix}$$

Page 537 new answers ——— 18 Jul 1995

19. The stated result holds, in fact, whenever the joint distribution of $U_0, \ldots, U_{n-1}$ is symmetrical (unchanged under permutations). Let $S_1 = U_0 + \cdots + U_{n-1}$, $S_2 = U_0^2 + \cdots + U_{n-1}^2$, $X = U_0U_1 + \cdots + U_{n-2}U_{n-1} + U_{n-1}U_0$, and $D = nS_2 - S_1^2$. Also let $\mathrm{E}\bigl(f(U_0, \ldots, U_{n-1})\bigr)$ denote the expected value of $f(U_0, \ldots, U_{n-1})$ subject to the condition $D \neq 0$. Since D is a symmetric function, $\mathrm{E}\bigl(f(U_0, \ldots, U_{n-1})\bigr) = \mathrm{E}\bigl(f(U_{p(0)}, \ldots, U_{p(n-1)})\bigr)$ for all permutations p of $\{0, \ldots, n-1\}$. Therefore we have $\mathrm{E}\, S_2/D = n\,\mathrm{E}\, U_0^2/D$, $\mathrm{E}\, S_1^2/D = n(n-1)\,\mathrm{E}(U_0U_1/D) + n\,\mathrm{E}\, U_0^2/D$, and $\mathrm{E}\, X/D = n\,\mathrm{E}(U_0U_1/D)$. It follows that $1 = \mathrm{E}\bigl((nS_2 - S_1^2)/D\bigr) = -(n-1)\,\mathrm{E}\bigl((nX - S_1^2)/D\bigr)$. (Strictly speaking, $\mathrm{E}\, S_2/D$ and $\mathrm{E}\, S_1^2/D$ might be infinite, so we should be careful to work only with linear combinations of expected values that are known to exist.)

20. Let E_{1111}, E_{211}, E_{22}, E_{31}, and E_4 denote the respective values $\mathrm{E}(U_0U_1U_2U_3/D^2)$, $\mathrm{E}(U_0^2U_1U_2/D^2)$, $\mathrm{E}(U_0^2U_1^2/D^2)$, $\mathrm{E}(U_0^3U_1/D^2)$, $\mathrm{E}(U_0^4/D^2)$. Then we have $\mathrm{E}\, S_2^2/D^2 = n(n-1)E_{22} + nE_4$, $\mathrm{E}(S_2S_1^2/D^2) = n(n-1)(n-2)E_{211} + n(n-1)E_{22} + 2n(n-1)E_{31} + nE_4$, $\mathrm{E}\, S_1^4/D^2 = n(n-1)(n-2)(n-3)E_{1111} + 6n(n-1)(n-2)E_{211} + 3n(n-1)E_{22} + 4n(n-1)E_{31} + nE_4$, $\mathrm{E}\, X^2/D^2 = n(n-3)E_{1111} + 2nE_{211} + nE_{22}$, $\mathrm{E}(XS_1^2/D^2) = n(n-2)(n-3)E_{1111} + 5n(n-2)E_{211} + 2nE_{22} + 2nE_{31}$, $\mathrm{E}\bigl((U_0 - U_1)^4/D^2\bigr) = 6E_{22} - 8E_{31} + 2E_4$, and the first result follows.

Let $\delta = \alpha\bigl((\ln n)/n\bigr)^{1/3}$, $M = \alpha^3/2 + 1/3$, and $m = \lceil 1/\delta \rceil$. If we divide the range of the distribution into m equiprobable parts, we can show that each part will contain between $n\delta(1-\delta)$ and $n\delta(1+\delta)$ points, with probability $\geq 1 - O(n^{-M})$, using the tail inequalities 1.2.9–(24) and (25). Hence, if the distribution is uniform, $D = \frac{1}{12}n^2\bigl(1 + O(\delta)\bigr)$ with at least this probability. If D is not in that range, we have $0 \leq (U_0 - U_1)^4/D^2 \leq 1$. Since $\mathrm{E}\bigl((U_0 - U_1)^4\bigr) = \int_0^1 \int_0^1 (x-y)^4\,dx\,dy = \frac{1}{15}$, we may conclude that $\mathrm{E}\bigl((U_0 - U_1)^4/D^2\bigr) = \frac{48}{5}n^{-4}\bigl(1 + O(\delta)\bigr) + O(n^{-M})$.

Note: Let N be the numerator of (23). When the variables all have the normal distribution, W. J. Dixon proved that the expected value of $e^{(wN+zD)/n}$ is

$$(1 - 2z - 2w)^{1/2}\bigl(1 - 2z + \sqrt{(1-2z)^2 - 4w^2}\,\bigr)^{-n/2} + O(w^n).$$

Differentiating with respect to w and integrating with respect to z, he found the moments $\mathrm{E}(N/D)^{2k-1} = (-\frac{1}{2})^{\overline{k}}/(n - \frac{1}{2})^{\overline{k}}$, $\mathrm{E}(N/D)^{2k} = (+\frac{1}{2})^{\overline{k}}/(n + \frac{1}{2})^{\overline{k}}$, when $n > 2k$. In particular, the variance in this case is exactly $1/(n+1) - 1/(n-1)^2$. [*Annals of Math. Stat.* **15** (1944), 119–144.]

Page 537 new answers ——— 20 Aug 1995

23. Let $P'(x_1, \ldots, x_t) = \frac{1}{\lambda'} \sum_{n=0}^{\lambda'-1} \bigl[(Y_n', \ldots, Y_{n+t-1}') = (x_1, \ldots, x_t)\bigr]$. Then we have

$$Q(x_1, \ldots, x_t) = \sum_{(y_1, \ldots, y_t)} P'(y_1, \ldots, y_t) P\bigl((x_1 - y_1) \bmod d, \ldots, (x_t - y_t) \bmod d\bigr)\,;$$

more compactly, $Q(x) = \sum_y P'(y)P(x-y)$. Hence, using the general inequality $(\mathrm{E}\, X)^2 \leq \mathrm{E}\, X^2$, we have $\sum_x \bigl(Q(x) - d^{-t}\bigr)^2 = \sum_x \bigl(\sum_y P'(y)\bigl(P(x-y) - d^{-t}\bigr)\bigr)^2 \leq \sum_x \sum_y P'(y)\bigl(P(x-y) - d^{-t}\bigr)^2 = \sum_y P'(y) \sum_x \bigl(P(x) - d^{-t}\bigr)^2 = \sum_x \bigl(P(x) - d^{-t}\bigr)^2$. [Cf. G. Marsaglia, *Comp. Sci. and Statistics: Symp. on the Interface* **16** (1984), 5–6. The result is of interest only when $d^t \leq 2\lambda$, since each $P(x)$ is a multiple of $1/\lambda$.]

24. Write $k : \alpha$ and $\alpha : k$ for the first k and last k elements of string α. Let $K(\alpha,\beta) = [\alpha=\beta]/P(\alpha)$, and let $\overline{C}$ be the $d^t \times d^t$ matrix with entries $\bar{c}_{\alpha\beta} = K(\alpha,\beta) - K(t-1 : \alpha, t-1 : \beta)$. Let C be the covariance matrix of the random variables $N(\alpha)$ for $|\alpha| = t$, divided by n. These variables are subject to the constraint $\sum_{a=0}^{d-1} N(\alpha a) = \sum_{a=0}^{d-1} N(a\alpha)$ for each of d^{t-1} strings α, but all other linear constraints are derivable from these (see Theorem 2.3.4.2G). Therefore C has rank $d^t - d^{t-1}$, and by exercise 3.3.1–25 it suffices to show that $C\overline{C}C = C$.

It is not difficult to verify that $c_{\alpha\beta} = P(\alpha\beta)\sum_{|k|<t} T_k(\alpha,\beta)$, where $T_k(\alpha,\beta)$ is a term corresponding to the overlap that might occur when we superimpose β on α and slide it k positions to the right:

$$T_k(\alpha,\beta) = \begin{cases} K(t+k : \alpha,\ \beta : t+k) - 1, & \text{if } k \le 0; \\ K(\alpha : t-k,\ t-k : \beta) - 1, & \text{if } k \ge 0. \end{cases}$$

For example, if $d = 2$, $t = 5$, $\alpha = 01101$, and $\beta = 10101$, we have $c_{\alpha\beta} = P(0)^4 P(1)^6 \times \bigl(P(01)^{-1} + P(101)^{-1} + P(1)^{-1} - 9\bigr)$. Entry $\alpha\beta$ of $C\overline{C}C$ is therefore $P(\alpha\beta)$ times

$$\sum_{|\gamma|=t-1}\ \sum_{a,b=0}^{d-1} P(\gamma ab) \sum_{|k|<t}\ \sum_{|l|<t} T_k(\alpha,\gamma a)\bigl(K(a,b)-1\bigr)T_l(\gamma b,\beta)\,.$$

Given k and l, the product $T_k(\alpha,\gamma a)\bigl(K(a,b)-1\bigr)T_l(\gamma b,\beta)$ expands to eight terms, each of which usually sums to ± 1 when multiplied by $P(\gamma ab)$ and summed over all γab. For example, the sum of $P(\gamma ab)K(2 : \alpha, \gamma a : 2)K(a,b)K(3 : \gamma b, \beta : 3)$, when $\alpha = a_1 \ldots a_t$, $\beta = b_1 \ldots b_t$, $\gamma = c_1 \ldots c_{t-1}$, and $t \ge 5$, is the sum of $P(c_4 \ldots c_{t-2})$, which is 1. If $t = 4$, the same sum would be $K(a_1, b_4)$, but it would cancel with the sum of $P(\gamma ab)K(2 : \alpha, \gamma a : 2)(-1)K(3 : \gamma b, \beta : 3)$. The net result is therefore 0 unless $k \le 0 \le l$; otherwise it turns out to be $K\bigl(i : (j : \alpha), (\beta : l) : i\bigr) - K\bigl(i-1 : (j : \alpha), (\beta : l+1) : i-1\bigr)$, where $i = \min(t+k, t-l)$ and $j = \max(0, k+l)$. The sum over k and l telescopes to $c_{\alpha\beta}$.

Page 537 new answer ________________ 17 Jul 1995

25. Empirical tests show, in fact, that when (22) is generalized to arbitrary t the ratios of corresponding elements of C_1^{-1} and $C_1^{-1}C_2C_1^{-1}$ are very nearly $-t$, when $t \ge 5$. For example, when $t = 6$ they all lie between -6.039 and -6.111; when $t = 20$ they all lie between -20.039 and -20.045. This phenomenon demands an explanation.

Page 537 new answers ________________ 06 Sep 1995

26. The vectors $(S_1, \ldots, S_n)$ are uniformly distributed points in the $(n-1)$-dimensional polyhedron defined by $S_1 \ge 0, \ldots, S_n \ge 0$ in the hyperplane $S_1 + \cdots + S_n = 1$. An easy induction proves that

$$\int_{s_1}^{\infty} dt_1 \int_{s_2}^{\infty} dt_2 \cdots \int_{s_{n-1}}^{\infty} dt_{n-1}\,[1 - t_1 - \cdots - t_{n-1} \ge s_n] = \frac{(1 - s_1 - s_2 - \cdots - s_n)_+^{n-1}}{(n-1)!}.$$

To get the probability, divide this integral by its value in the special case $s_1 = \cdots = s_n = 0$. [Bruno de Finetti, *Giornale Istituto Italiano degli Attuari* **27** (1964), 151–173.]

(b) The probability that $S_{(1)} \ge s$ is the probability that $S_1 \ge s, \ldots, S_n \ge s$.

(c) The probability that $S_{(k)} \ge s$ is the probability that at most $k-1$ of the S_j are $< s$; hence $1 - F_k(s) = G_1(s) + \cdots + G_{k-1}(s)$, where $G_j(s)$ is the probability that exactly j spacings are $< s$. By symmetry, $G_j(s)$ is $\binom{n}{j}$ times the probability that $S_1 < s, \ldots, S_j < s, S_{j+1} \ge s, \ldots, S_n \ge s$; and the latter is $\Pr(S_1 < s, \ldots, S_{j-1} < s,$

$S_j \geq 0, S_{j+1} \geq s, \ldots, S_n \geq s) - \Pr(S_1 < s, \ldots, S_{j-1} < s, S_j \geq s, \ldots, S_n \geq s)$. Repeated application of (a) shows that $G_j(s) = \binom{n}{j}\sum_l \binom{j}{l}(-1)^{j-l}(1-(n-l)s)_+^{n-1}$; hence

$$1 - F_k(s) = \sum_l \binom{n}{l}\binom{n-l-1}{k-l-1}(-1)^{k-l-1}(1-(n-l)s)_+^{n-1}.$$

In particular, the largest spacing $S_{(n)}$ has distribution

$$F_n(s) = 1 - \sum_l \binom{n}{l}\binom{n-l-1}{n-l-1}(-1)^{n-l-1}(1-(n-l)s)_+^{n-1} = \sum_l \binom{n}{l}(-1)^l(1-ls)_+^{n-1}.$$

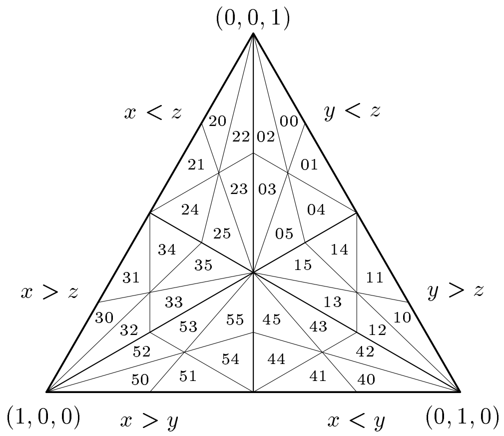
[Incidentally, the similar quantity $x^{n-1}(n-1)!^{-1}F_n(x^{-1})$ turns out to be the *density* function for the *sum* $U_1 + \cdots + U_n$ of uniform deviates.]

(d) From the formulas $\mathrm{E}\, s^r = r\int_0^1 (1-F(s))s^{r-1}\,ds$ and $\int_0^1 s^r(1-ks)_+^{n-1}\,ds = k^{-r-1}n^{-1}\binom{n+r}{r}^{-1}$, we find $\mathrm{E}\, S_{(k)} = n^{-1}(H_n - H_{n-k})$ and, with a bit of algebra, $\mathrm{E}\, S_{(k)}^2 = n^{-1}(n+1)^{-1}(H_n^{(2)} - H_{n-k}^{(2)} + (H_n - H_{n-k})^2)$. Thus the variance of $S_{(k)}$ is $n^{-1}(n+1)^{-1}(H_n^{(2)} - H_{n-k}^{(2)} - (H_n - H_{n-k})^2/n)$.

[The distributions $F_k(s)$ were first found by W. A. Whitworth, who stated the question as problem 667 in *Choice and Chance* (Cambridge, 1867) and gave the solution in *DCC Exercises in Choice and Chance* (Cambridge, 1897). Whitworth also discovered an elegant way to compute the expected value of any polynomial in the functions $G_k(s) = F_k(s) - F_{k+1}(s)$; this was published in a booklet entitled *The Expectation of Parts* (Cambridge, 1898), and incorporated into the fifth edition of *Choice and Chance* (1901). Simplified expressions for the mean and variance and for a variety of more general spacing statistics were found by Barton and David, *J. Royal Stat. Soc.* **B18** (1956), 79–94. See R. Pyke, *J. Royal Stat. Soc.* **B27** (1965), 395–449, for a survey of the ways in which statisticians have traditionally analyzed spacings as clues to potential biases in data.]

27. Consider the polyhedron in the hyperplane $S_1 + \cdots + S_n = 1$ defined by the inequalities $S_1 \geq 0, \ldots, S_n \geq 0$. This polyhedron consists of $n!$ congruent subpolyhedra defined by the ordering of the S's (assuming that the S's are distinct), and the operation of sorting is an $n!$-to-1 folding of the large polyhedron to the subpolyhedron in which $S_1 \leq \cdots \leq S_n$. The transformation that takes $(S_{(1)}, \ldots, S_{(n)})$ to $(S_1', \ldots, S_n')$ is a 1-to-1 mapping that expands differential volumes by the factor $n!$. It takes the vertices $(\frac{1}{n}, \ldots, \frac{1}{n})$, $(0, \frac{1}{n-1}, \ldots, \frac{1}{n-1})$, $\ldots$, $(0, \ldots, 0, 1)$ of the subpolyhedron into the respective vertices $(1, 0, \ldots, 0)$, $(0, 1, 0, \ldots, 0)$, $\ldots$, $(0, \ldots, 0, 1)$, linearly stretching and distorting the overall shape in the process. (The Euclidean distances between vertices $(0, \ldots, 0, \frac{1}{j}, \ldots, \frac{1}{j})$ and $(0, \ldots, 0, \frac{1}{k}, \ldots, \frac{1}{k})$ in the subpolyhedron are $|j^{-1} - k^{-1}|^{1/2}$, but the transformation produces a regular simplex in which all n vertices are $\sqrt{2}$ apart.)

The behavior of iterated spacings is easiest to understand if we examine the details graphically when $n = 3$. In this case the polyhedron is simply an equilateral triangle, whose points are represented with barycentric coordinates (x, y, z), $x + y + z = 1$. The accompanying diagram illustrates the first two levels of a recursive decomposition of this triangle. Each of the

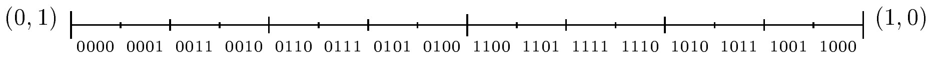


6^2 subtriangles has been labeled with a two-digit code pq, where p represents the applicable permutation when $(x, y, z) = (S_1, S_2, S_3)$ is sorted into $(S_{(1)}, S_{(2)}, S_{(3)})$, and q represents the permutation in the next stage when S'_1, S'_2, and S'_3 are sorted, according to the following code:

$$0\!: x<y<z,\quad 1\!: x<z<y,\quad 2\!: y<x<z,\quad 3\!: y<z<x,\quad 4\!: z<x<y,\quad 5\!: z<y<x.$$

For example, the points of subtriangle 34 have $S_2 < S_3 < S_1$ and $S'_3 < S'_1 < S'_2$. We can continue this process to infinitely many levels; all points of the triangle with irrational barycentric coordinates thereby acquire a unique representation as an infinite radix-6 expansion. A tetrahedron can be subdivided similarly into 24, 24^2, 24^3, … subtetrahedra, and in general this procedure constructs a radix-$n!$ expansion for the points of any $(n-1)$-dimensional simplex.

When $n = 2$ the process is especially simple: If $x \notin \{0, \frac{1}{2}, 1\}$, the transformation takes spacings $(x, 1-x) = (x, y)$ into either $(2x \bmod 1, 2y \bmod 1)$ or $(2y \bmod 1, 2x \bmod 1)$, depending on whether $x < y$ or $x > y$. Repeated tests therefore essentially shift the binary representation left one bit, possibly complementing the result. After at most $e+1$ iterations on e-bit numbers the process must converge to the fixed point $(0, 1)$. Permutation coding in the case $n = 2$ corresponds simply to folding and stretching a line; the first four levels of subdivision have the following four-bit codes:



This sequence is exactly the Gray binary code studied in Section 7.2.1. In general, the radix-$n!$ permutation code for an n-simplex has the property that adjacent regions have identical codes except in one digit position. Each iteration of the spacing transformation shifts off the leftmost digit of the representation of each point. Note that equal birthday spacings are points near the boundary of the first-level decomposition.

This fundamental transformation from $(S_1, \ldots, S_n)$ to $(S'_1, \ldots, S'_n)$ is implicit in Whitworth's proof of Proposition LVI in the fifth edition of *Choice and Chance* (see the reference in answer 26). It was first studied explicitly by J. Durbin [*Biometrika* **48** (1961), 41–55], who was inspired by a similar construction of P. V. Sukhatme [*Annals of Eugenics* **8** (1937), 52–56]. The permutation coding for iterated spacings was introduced by H. E. Daniels [*Biometrika* **49** (1962), 139–149].

28. (a) The number of partitions of m into n distinct positive parts is $p_n(m - \binom{n+1}{2})$, by exercise 5.1.1–16. These partitions can be permuted in $n!$ ways to yield n-tuples $(y_1, \ldots, y_n)$ with $0 = y_1 < y_2 < \cdots < y_n < m$; and each of these n-tuples leads to $(n-1)!$ n-tuples that have $y_1 = 0$ and $0 < y_2, \ldots, y_n < m$. Now add a constant mod m to each y_j; this preserves the spacings. Hence $b_{n00}(m) = m\, n!\, (n-1)!\, p_n(m - \binom{n+1}{2})$.

(b) Zero spacings correspond to balls in the same urn, and they contribute $s - 1$ to the count of equal spacings. Therefore $b_{nrs}(m) = \left\{ {n \atop n-s} \right\} b_{(n-s)(r+1-s)0}(m)$.

(c) Since $\left\{ {n \atop n-1} \right\} = \binom{n}{2}$, the probability is $n!\,(n-1)!\, m^{1-n}\bigl(p_n(m - \binom{n+1}{2}) - \frac{1}{2}p_{n-1}(m - \binom{n}{2})\bigr)$.

29. By the previous answer and exercise 5.1.1–15 we have $b_{n0}(z) = n!\,(n-1)!\, z^{\binom{n+1}{2}} / (1-z)\ldots(1-z^n)$. When $r = 1$, the $n!$ in our previous derivation becomes $n!/2$, and the number of solutions to $0 < s_1 < \cdots < s_k \le s_{k+1} < \cdots < s_n$ with $s_1 + \cdots + s_n = m$

is the number of solutions to $0 \le s_1 - 1 \le \cdots \le s_k - k \le s_{k+1} - k \le \cdots \le s_n - n + 1$ with $(s_1 - 1) + \cdots + (s_k - k) + (s_{k+1} - k) + \cdots + (s_n - n + 1) = m - \binom{n}{2} - k$. Hence $b_{n1}(z) = \frac{1}{2} n!\,(n-1)! \sum_{k=1}^{n} (z^k - z^n)\, z^{\binom{n}{2}}/(1-z)\ldots(1-z^n)$. A similar argument shows that

$$\frac{b_{n2}(z)}{n!\,(n-1)!} = \Biggl(\frac{1}{2!\,2!} \sum_{1\le j<k<n} (z^j - z^n)(z^k - z^{n-1}) \\ + \frac{1}{3!} \sum_{1\le k<n} (z^k - z^n)(z^k - z^{n-1})\Biggr) \frac{z^{\binom{n-1}{2}}}{(1-z)\ldots(1-z^n)}.$$

We can obtain $b_{nr}(z)$ for general r from the formula

$$\frac{\sum_r b_{nr}(z) w^r}{n!\,(n-1)!\,z^n} = \sum_{0\le b_1,\ldots,b_{n-1}\le 1} \frac{(z - b_1 z^n)\ldots(z^{n-1} - b_{n-1} z^n)}{c_1 \ldots c_{n-1}(1-z)\ldots(1-z^n)} \Bigl(\frac{w}{z^{n-1}}\Bigr)^{b_1} \cdots \Bigl(\frac{w}{z^1}\Bigr)^{b_{n-1}}$$

where $c_k = 1 + b_k + b_k b_{k-1} + \cdots + b_k \ldots b_2 b_1 = 1 + b_k c_{k-1}$. (The special case $w = 1$ is interesting because the left side sums to $(1-z)^{-n}/n!$ in that case.)

30. This is a good problem for the saddle point method [N. G. de Bruijn, *Asymptotic Methods in Analysis* (North-Holland, 1961), Chapter 5]. We have $p_n(m) = \frac{1}{2\pi i} \oint e^{f(z)} \frac{dz}{z}$, where $f(z) = -m \ln z - \sum_{k=1}^{n} \ln(1 - z^k)$. Let $\rho = n/m$ and $\delta = \sqrt{n}/m$; integrating on the path $z = e^{-\rho + it\delta}$ gives $p_n(m) = \frac{\delta}{2\pi} \int_{-\pi/\delta}^{\pi/\delta} \exp\bigl(f(e^{-\rho+it\delta})\bigr)\, dt$. It is convenient to use the identity

$$g(se^t) = \sum_{j=0}^{n} \frac{t^j}{j!} \vartheta^j g(s) + \int_0^t \frac{u^n}{n!} \vartheta^{n+1} g(se^{t-u})\, du\,,$$

where $g = g(z)$ is any analytic function and ϑ is the operator $z\frac{d}{dz}$. When the function $\vartheta^j g$ is evaluated at e^z the result is the same as when $g(e^z)$ is differentiated j times with respect to z. This principle leads to the formula

$$\vartheta^j f(e^{-\rho}) = -m[j=1] + \frac{j!\,n}{\rho^j} + (-1)^j \sum_{k=1}^{n} \sum_{l\ge j} \frac{l^{\underline{j}} B_l}{l \cdot l!} k^l \rho^{l-j},$$

because of another handy identity,

$$\ln\Bigl(\frac{1 - e^{-z}}{z}\Bigr) = \sum_{n\ge 1} \frac{B_n z^n}{n \cdot n!}\,.$$

Therefore we obtain an asymptotic expansion of the integrand,

$$\begin{aligned}\exp\bigl(f(e^{-\rho+it\delta})\bigr) &= \exp\Biggl(\sum_{j\ge 0} \frac{i^j \delta^j t^j}{j!} \vartheta^j f(e^{-\rho})\Biggr)\\ &= e^{-t^2/2 + f(e^{-\rho})} \exp(ic_1 t - c_2 t^2 - ic^3 t^3 + \cdots),\end{aligned}$$

where $c_1 = \left(\frac{n(n+1)}{2}B_1 + \frac{n(n+1/2)(n+1)}{6}B_2\rho\right)\delta + O(n^{-3})$, etc.; and it turns out that $c_j = O(n^{-3})$ for $j \ge 8$. Factoring out the constant term

$$\begin{aligned}\frac{\delta}{2\pi}e^{f(e^{-\rho})} &= \frac{\delta}{2\pi\, n!\,\rho^n e^{-m\rho}}\exp\left(-\sum_{k=1}^{n}\sum_{l\ge 1}\frac{B_l}{l\cdot l!}k^l\rho^l\right)\\ &= \frac{\sqrt{n}\,m^{n-1}e^{n+\alpha/4}}{2\pi\,n!\,n^n}\left(1+\frac{18\alpha-\alpha^2}{72n}+\frac{108\alpha^2-36\alpha^3+\alpha^4}{10368n^2}+O(n^{-3})\right)\end{aligned}$$

leaves us with an integral whose integrand is exponentially small when $|t| \ge n^{\epsilon}$. We can ignore larger values of t, because partial fraction expansion shows that the integrand is $O\bigl((m/n)^{n/2}\bigr)$; none of the other roots of unity occurs more than $n/2$ times as a pole of the denominator. Hence we are allowed to "trade tails" [*CMath*, §9.4] and integrate over all t. The formulas $\int_{-\infty}^{\infty} e^{-t^2/2}t^j\,dt = (j-1)(j-3)\ldots(1)\sqrt{2\pi}\,[j \text{ even}]$ and $n! = (n/e)^n\sqrt{2\pi n}\exp\bigl(\frac{1}{12}n^{-1} + O(n^{-3})\bigr)$ suffice to complete the evaluation.

With $q_n(m) = p_n\bigl(m - \binom{n+1}{2}\bigr)$ in place of $p_n(m)$ the calculation proceeds in the same way but with c_1 increased by $\frac{1}{2}\alpha(n^{1/2} - n^{-1/2})$ and with the additional factor $\exp\bigl(-\rho\binom{n+1}{2}\bigr)$. We get

$$q_n(m) = \frac{m^{n-1}e^{-\alpha/4}}{n!\,(n-1)!}\left(1 - \frac{13\alpha^2}{288n} + \frac{169\alpha^4 - 2016\alpha^3 - 1728\alpha^2 + 41472\alpha}{165888n^2} + O(n^{-3})\right);$$

this matches the formula for $p_n(m)$ except that α has been changed to $-\alpha$. (In fact, if we define $p_n(m) = r_n\bigl(2m + \binom{n+1}{2}\bigr)$ and $q_n(m) = r_n\bigl(2m - \binom{n+1}{2}\bigr)$, the generating function $R_n(z) = \sum_m r_n(z^m) = \prod_{k=1}^{n}(z^{-k} - z^k)^{-1}$ satisfies $R_n(1/z) = (-1)^n R_n(z)$. This implies a duality formula $r_n(-m) = (-1)^{n-1}r_n(m)$, in the sense that this equation is identically true when we express $r_n(m)$ as a polynomial in m and roots of unity. Therefore we may say that $q_n(m) = p_n(-m)$. A general treatment of such duality can be found in G. Pólya, *Math. Zeitschrift* **29** (1928), 549–640, §44.) For further information see G. Szekeres, *Quarterly J. Math. Oxford* **2** (1951), 85–108; **4** (1093), 96–111.

The exact value of $q_n(m)$ when $m = 2^{25}$ and $n = 512$ is $7.08069\,34695\,90264\,094\ldots \times 10^{1514}$; our approximation gives the estimate $7.080693501 \times 10^{1514}$.

The probability that the birthday test finds $R = 0$ spacings is $b_{n00}(m)/m^n = n!\,(n-1)!\,m^{1-n}q_n(m) = e^{-\alpha/4} + O(n^{-1})$, by exercise 28, because the contribution from $b_{n01}(m)$ is $\approx \frac{\alpha}{2n}e^{-\alpha/4} = O(n^{-1})$. Inserting the factor $g_n(z) = \sum_{k=1}^{n-1}(z^{-k} - 1)$ into the integrand for $q_n(m)$ has the effect of multiplying the result by $\frac{\alpha}{2} + O(n^{-1})$, because $g_n(e^{-\rho+it\delta}) = \binom{n}{2}\rho + O(n^3\rho^2) + itO(n^2\delta) - \frac{1}{2}t^2O(n^3\delta^2) + \cdots$. Similarly, the extra factor $\sum_{1\le j<k<n}(z^{-j} - 1)(z^{-k} - 1)$ essentially multiplies by $\frac{1}{8}n^4\rho^2 = \frac{1}{8}\alpha^2$, plus $O(n^{-1})$; other contributions to the probability that $R = 2$ are $O(n^{-1})$. In this way we find that the probability of r equal spacings is $e^{-\alpha/4}(\alpha/4)^r/r! + O(n^{-1})$, a Poisson distribution; more complicated terms arise if we carry the expansion out to $O(n^{-2})$.

Page 537 new answers ______ 23 Sep 1995

31. The 79 bits consist of 24 sets of three, $\{Y_n, Y_{n+31}, Y_{n+55}\}$, $\{Y_{n+1}, Y_{n+32}, Y_{n+56}\}$, $\ldots$, $\{Y_{n+23}, Y_{n+54}, Y_{n+78}\}$, plus 7 additional bits $Y_{n+24}, \ldots, Y_{n+30}$. The latter bits are equally likely to be 0 or 1, but in each group of three the probability is $\frac{1}{4}$ that the bits will be $\{0,0,0\}$ and $\frac{3}{4}$ that they will be $\{0,1,1\}$. Therefore the probability generating function for the sum of bits is $f(z) = \bigl(\frac{1+z}{2}\bigr)^7\bigl(\frac{1+3z^2}{4}\bigr)^{24}$, a polynomial of degree 55. (Well, not quite; strictly speaking, it is $(2^{55}f(z) - 1)/(2^{55} - 1)$, because the all-0 case is excluded.) The coefficients of $2^{55}f(z)$ are easily computed by machine, and we find that the probability of more 1s than 0s is $18509401282464000/(2^{55}-1) \approx 0.51374$.

Notes: This exercise is based on the discovery by Vattulainen, Ala-Nissila, and Kankaala [*Physical Review Letters* **73** (1994), 2513–2516] that a lagged Fibonacci generator fails a more complicated two-dimensional random walk test. Notice that the sequence Y_{2n}, Y_{2n+2}, $\ldots$ will fail the test too, because it satisfies the same recurrence. The bias toward 1s also carries over into the subsequence consisting of the even-valued elements generated by $X_n = (X_{n-55} \pm X_{n-24}) \bmod 2^e$; we tend to have more occurrences of $(\ldots 10)_2$ than $(\ldots 00)_2$ in binary notation.

There's nothing magic about the number 79 in this test; experiments show that a significant bias towards a majority of 1s is present also in random walks of length 101 or 1001 or 10001. But a formal proof seems to be difficult. After 86 steps the generating function is $(\frac{1+3z^2}{4})^{17}(\frac{1+2z^2+4z^3+z^4}{8})^7$; then we get the factors $(1+2z^2+5z^3+5z^4+10z^5+8z^6+z^7)/32$; then $(1+2z^2+7z^3+7z^4+15z^5+25z^6+29z^7+28z^8+13z^9+z^{10})/128$, etc. The analysis becomes more and more complicated as the walks get longer.

Intuitively, the preponderance of 1s that arise in the first 79 steps ought to persist as long as the subsequent numbers are reasonably balanced between 0 and 1. The accompanying diagram shows the results of a much smaller case, the generator $Y_n = (Y_{n-2}+Y_{n-11}) \bmod 2$, which is easy to analyze exhaustively. In this case random walks of length 445 have a 64% chance of finishing to the right of the starting point; this bias disappears only when the length of the walk increases to half the period length (after which, of course, 0s are more likely, although the full period does lack one 0).

The probability that 1s outnumber 0s in random m-tuples when $Y_n = Y_{n-2} \oplus Y_{n-11}$.

Lüscher's discarding technique can be used to avoid the bias toward 1s (see the end of Section 3.2.2). For example, with lags 55 and 24, no deviation for randomness is observed for random walks of length 1001 when the numbers are generated in batches of 165, if only the first 55 numbers of each batch are used.

33. Not if they take the values $(-1-2\epsilon^2, \epsilon-2\epsilon^2, 1-2\epsilon)$ with respective probabilities $(\frac{1}{2}-\epsilon, \epsilon, \frac{1}{2})$. Then $X+Y>0$ with probability $(\frac{1}{2}+\epsilon)^2 < \frac{1}{2}$, if ϵ is sufficiently small. [Thus, two golfers might be of equal strength based on their mean scores, but one might be more likely to win a one-round tournament while the other will more often win in two rounds. See T. M. Cover, *Amer. Statistician* **43** (1989), 277-278, for a discussion of similar phenomena.]

34. We essentially want $[z^{(k+l-1)/2}]\,(\frac{1+z}{2})^{k-2l}(\frac{1+3z^2}{4})^l/(1-z)$. Let $m=k-2l$ and $n=l$; the desired coefficient is $\frac{1}{2\pi i}\oint e^{g(z)}\frac{dz}{z(1-z)}$, where $g(z) = m\ln(\frac{1+z}{2})+n\ln(\frac{1+3z^2}{4})-(\frac{m+3n-1}{2})\ln z$. It is convenient (and saddle-wise) to integrate along the path $z=e^{\epsilon u}$ where $\epsilon^2 = 4/(m+3n)$ and $u=-1+it$ for $-\infty < t < \infty$. We have $g(e^{\epsilon u}) = -\frac{1}{2}\epsilon u + u^2/2 + c_3\epsilon u^3 + c_4\epsilon^2u^4 + \cdots$, where $c_k = \epsilon^2\vartheta^k g(1)/k! = O(1)$. Also $1/(1-e^{\epsilon u}) = \frac{-1}{\epsilon u} + \frac{1}{2} - B_2\epsilon u/2! - \cdots$. Multiplying out the integrand and using the facts that $\frac{1}{2\pi i}\int_{1-i\infty}^{1+i\infty} e^{u^2/2}\,\frac{du}{u} = \frac{1}{2}$ and $\frac{1}{2\pi i}\int_{a-i\infty}^{a+i\infty} e^{u^2/2}u^{2k}\,du = (-1)^k(2k-1)(2k-3)\ldots(1)\sqrt{2\pi}$ yields the asymptotic formula $\frac{1}{2} + (2\pi)^{-1/2}n(m+3n)^{-3/2} + O\bigl((m+3n)^{-3/2}\bigr)$. If

$m+3n$ is even, the same asymptotic formula holds, provided that we give half of the coefficient of $z^{(m+3n)/2}$ to the 1s and half to the 0s. (This coefficient is $(\frac{2}{\pi(m+3n)})^{1/2} + O((m-3n)^{-3/2})$.)

Page 537 new answer ——— 06 Jan 1997

32. (a) $\mathrm{E}\,S_m = N^{-1}\sum_{n=0}^{N-1}\sum_{j=0}^{m-1} Z_{n+j} = N^{-1}\sum_{j=0}^{m-1}\sum_{n=0}^{N-1} Z_{n+j} = m/N$, because $\sum_{n=0}^{N-1} Z_{n+j} = 2^{k-1} - (2^{k-1}-1) = 1$.

(b) Let $\xi^k = a_1\xi^{k-1} + \cdots + a_k$, and define the linear function f as in the first solution to exercise 3.2.2–16. Then $Y_n = f(\xi^n)$, and it follows that $Y_{n+i} + Y_{n+j} = f(\xi^{n+i}) + f(\xi^{n+j}) \equiv f(\xi^{n+i}+\xi^{n+j}) = f(\xi^n\alpha)$ (modulo 2), where α is nonzero when $i \not\equiv j$ (modulo N). Hence $\mathrm{E}\,S_m^2 = N^{-1}\sum_{i=0}^{m-1}\sum_{j=0}^{m-1}\sum_{n=0}^{N-1} Z_{n+i}Z_{n+j} = N^{-1}(\sum_{i=0}^{m-1}\sum_{n=0}^{N-1} Z_{n+i}^2 - 2\sum_{0\le i<j<m}\sum_{n=0}^{N-1} Z_n) = m - m(m-1)/N$.

(c) $\mathrm{E}\sum_{j=0}^{m-1} Z_{n+j} = \sum_{j=0}^{m-1}\mathrm{E}\,Z_{n+j} = 0$ and $\mathrm{E}(\sum_{j=0}^{m-1} Z_{n+j})^2 = \sum_{j=0}^{m-1}\mathrm{E}\,Z_{n+j}^2 + \sum_{0\le i<j<m}(\mathrm{E}\,Z_{n+i})(\mathrm{E}\,Z_{n+j}) = m$ when each Z_n is truly random. Thus the mean and variance of S_m are very close to the correct values when $m \ll N$.

(d) $\mathrm{E}\,S_m^3 = N^{-1}\sum_{h=0}^{m-1}\sum_{i=0}^{m-1}\sum_{j=0}^{m-1}\sum_{n=0}^{N-1} Z_{n+h}Z_{n+i}Z_{n+j}$. If any of h, i, or j are equal, the sum on n is 1; hence

$$\mathrm{E}\,S_m^3 = \frac{1}{N}\left(m^3 - m^{\underline{3}} + 6\sum_{0\le h<i<j<m}\sum_{n=0}^{N-1} Z_{n+h}Z_{n+i}Z_{n+j}\right).$$

Arguing as in (b), we find that the sum on n will be 1 if $\xi^h+\xi^i+\xi^j \ne 0$; otherwise it will be $-N$. Thus $\mathrm{E}\,S_m^3 = m^3 - 6B(N+1)/N$, where $B = \sum_{0\le h<i<j<m}[\xi^h+\xi^i+\xi^j=0] = \sum_{0<i<j<m}[1+\xi^i+\xi^j=0]\,(m-j)$. Finally observe that $1+\xi^i=\xi^j$ in the field if and only if $f(\xi^{i+l}) = f(\xi^{j+l})$ for $0<l<k$, assuming that $0<i<j<N$.

(e) The only nonzero term occurs for $i=31$ and $j=55$; hence $B = 79-55 = 24$. (The next nonzero term occurs when $i=62$ and $j=110$.) In a truly random situation, $\mathrm{E}\,S_m^3$ should be zero, so this value $\mathrm{E}\,S_{79}^3 \approx -144$ is distinctly nonrandom. Curiously it is negative, although the previous exercise showed that S_{79} is usually *positive.* The value of S_{79} tends to be more seriously negative when it does dip below zero.

Reference: IEEE Trans. **IT-14** (1968), 569–576. Experiments by M. Matsumoto and Y. Kurita [*ACM Trans. Modeling and Comp. Simul.* **2** (1992), 179–194; **4** (1994), 254–266] confirm that trinomial-based generators fail such distribution tests even when the lags are quite large.

Page 537 new answer ——— 01 Oct 1995

35. The number of strings of length n that exclude a given two-letter substring or pair of substrings is the coefficient of z^n in an appropriate generating function, and it can be written $ce^{n\tau}m^n + O(1)$ where c and τ have series expansions in powers of $\epsilon = 1/m$:

Case	Excluded	Generating function	c	τ
1	aa	$(1+z)/p(z)$	$1+\epsilon^2-2\epsilon^3+\cdots$	$-\epsilon^2+\epsilon^3-\frac{5}{2}\epsilon^4+\cdots$
2	ab	$1/(1-mz+z^2)$	$1+\epsilon^2+3\epsilon^4+\cdots$	$-\epsilon^2-\frac{3}{2}\epsilon^4+\cdots$
3	aa, bb	$(1+z)/(p(z)+z^2)$	$1+2\epsilon^2-4\epsilon^3+\cdots$	$-2\epsilon^2+2\epsilon^3-8\epsilon^4+\cdots$
4	aa, bc	$(1+z)/(p(z)+z^2+z^3)$	$1+2\epsilon^2-2\epsilon^3+\cdots$	$-2\epsilon^2+\epsilon^3-7\epsilon^4+\cdots$
5	ab, bc	$(1+z)/(1-mz+2z^2-z^3)$	$1+2\epsilon^2-2\epsilon^3+\cdots$	$-2\epsilon^2+\epsilon^3-6\epsilon^4+\cdots$
6	ab, cd	$1/(1-mz+2z^2)$	$1+2\epsilon^2+12\epsilon^4+\cdots$	$-2\epsilon^2-6\epsilon^4+\cdots$

(Here a,b,c,d denote distinct letters and $p(z) = 1-(m-1)(z+z^2)$. It turns out that the effect of excluding $\{ab,ba\}$ or $\{aa,ab\}$ is equivalent to excluding $\{aa,bb\}$; excluding

$\{ab, ac\}$ is equivalent to excluding $\{ab, cd\}$.) Let $S_n^{(j)}$ be the coefficient of z^n in Case j and let X be the total number of two-letter combinations that do not appear. Then $\mathrm{E}\,X = (mS_n^{(1)} + m^{\underline{2}}S_n^{(2)})/m^n$ and $\mathrm{E}\,X^2 = (mS_n^{(1)} + m^{\underline{2}}(S_n^{(2)} + 6S_n^{(3)}) + 2m^{\underline{3}}(S_n^{(4)} + S_n^{(5)} + S_n^{(6)}) + m^{\underline{4}}S_n^{(6)})/m^n$.

▸**Page 537** line 1 of answer 3 ——— 06 May 1995

$f(x) = \sum_{n\ge1}(-\sin 2\pi nx)/n \leadsto ((x)) = -\sum_{n\ge1} \frac{1}{n\pi} \sin 2\pi nx$

▸**Page 538** answer 17 ——— 18 Dec 1982

line 1: $p_j - 1 \leadsto p_{j-1}$
line 6: $A \leftarrow A - (a-6b)s \leadsto A \leftarrow A + (a-6b)s$

▸**Page 539** line 4 of answer 20 ——— 07 May 1997

$s(x'') \leadsto s(s(x''))$

▸**Page 540** line 3 ——— 02 Feb 1981

$$\left(\left(\frac{a(x+c_0/d}{m/d}\right)\right) \leadsto \left(\left(\frac{a(x+c_0/d)}{m/d}\right)\right)$$

▸**Page 543** line 5 of answer 5 ——— 02 Feb 1981

$(h' - qh)^2 \leadsto (h' - q'h)^2$

▸**Page 544** replacement for lines 3–5 ——— 16 Jun 1997

Choosing S so that its first column is a multiple of U_1^T and its other columns are any suitable vectors, we have

$$US = \begin{pmatrix} \alpha_1 & 0 & \ldots & 0 \\ \alpha_2 & & & \\ \vdots & & U' & \\ \alpha_t & & & \end{pmatrix}$$

Page 544 line -4 ——— 27 Sep 1996

(modulo p).] $\leadsto$ (modulo p). The values of u and v also appear when Euclid's algorithm for integers is applied in the ordinary way to p and x; cf. J. A. Serret and C. Hermite, *J. Math. Pures et Appl.* **5** (1848), 12–15.]

▸**Page 545** broken type at end of line 2 ——— 29 Aug 1995

(modulo 10).

Page 545 replacement for lines 8–13 ——— 29 Aug 1995

Line 14 of the table was obtained as follows: $\lfloor 2^{32}/\sqrt{3}\rfloor = 2479700524$; we drop down to $N = 2479700521$, which equals $37 \cdot 797 \cdot 84089$ and has four solutions $N = 4364^2 + 49605^2 = 26364^2 + 42245^2 = 38640^2 + 31411^2 = 11960^2 + 48339^2$. The corresponding multipliers are 2974037721, 2254986297, 4246248609, and 956772177. We try also $N - 4$, but it is ineligible because it is divisible by 3. On the other hand the prime number $N - 8 = 45088^2 + 21137^2$ leads to the multiplier 3825140801. Similarly, we get additional multipliers from $N-20$, $N-44$, $N-48$, etc. The multiplier on line 14 is the best of the first sixteen multipliers found by this procedure; it's one of the four obtained from $N - 68$.

Page 546 replacement for answer 20 ——— 11 Aug 1995

20. When $a \bmod 8 = 5$, the points $2^{-e}\bigl(x, s(x), \ldots, s^{t-1}(x)\bigr)$ for x in the period are the same as the points $2^{2-e}\bigl(y, \sigma(y), \ldots, \sigma^{t-1}(y)\bigr)$ for $0 \le y < 2^{e-2}$, plus $(X_0 \bmod 4)/2^e$, where $\sigma(y) = \bigl(ay + \lfloor a/4 \rfloor\bigr) \bmod 2^{e-2}$. So in this case we should use Algorithm S with $m = 2^{e-2}$.

When $a \bmod 8 = 3$, the maximum distance between parallel hyperplanes that cover the points $2^{-e}\bigl(x, s(x), \ldots, s^{t-1}(x)\bigr)$ modulo 1 is the same as the maximum distance covering the points $2^{-e}\bigl(x, -s(x), \ldots, (-1)^{t-1}s^{t-1}(x)\bigr)$, because the negation of coordinates doesn't change distance. The latter points are $2^{2-e}\bigl(y, \sigma(y), \ldots, \sigma^{t-1}(y)\bigr)$ where $\sigma(y) = \bigl(-ay - \lceil a/4 \rceil\bigr) \bmod 2^{e-2}$, plus a constant offset. Again we apply Algorithm S with $m = 2^{e-2}$; changing a to $m - a$ has no effect on the result.

▸**Page 546** line 2 of answer 24 ——— 02 Feb 1981

mod n ⤳ mod m

Page 547 line 8 ——— 27 Aug 1995

T225.] ⤳ T225; L'Ecuyer, Blouin, and Couture, *ACM Trans. Modeling and Comput. Sim.* **3** (1993), 87–98.]

▸**Page 547** line 7 of answer 27 ——— 08 May 1997

maximum ⤳ minimum
(also change $\rho_{\max}$ ⤳ $\rho_{\min}$ throughout that answer)

▸**Page 547** line 10 of answer 27 ——— 02 Feb 1981

$r(u_1, \ldots, r_t)$ ⤳ $r(u_1, \ldots, u_t)$

▸**Page 548** answer 30 ——— 11 Aug 1995

line 1: We wish to minimize ⤳ First let's minimize
line 8: $Q_{s-n-1}(n+2, \ldots, a_s)$ ⤳ $Q_{s-n-1}(a_{n+2}, \ldots, a_s)$
new paragraph to follow line 9:

Now let $f(u) = m \sin(\pi u/m)$ and $A = \max(a_1, \ldots, a_s)$. We have $2u \le f(u) \le \pi u$ for $0 \le u \le \frac{1}{2}m$, and $f(m-u) = f(u)$. Hence we can assume that $r_{\max} = 1/f(u)f(au \bmod m)$ for some u with $1 \le u \le \frac{1}{2}m$. Setting $u' = \min(au \bmod m, (-au) \bmod m)$, we have $r_{\max} = 1/f(u)f(u')$. We know from the previous paragraph that $uu' \ge qq'$, where $A/m \le 1/qq' \le (A+2)/m$. Hence $f(u)f(u') \ge 4uu' \ge 4qq'$, and we have $r_{\max} \le (A+2)/(4m)$. (There is a similar lower bound, since $r_{\max} \ge 1/f(q)f(q') \ge A/(\pi^2 m)$.)

Page 548 line −3 ——— 20 Mar 1995

to appear ⤳ *BIT* **25** (1980), 193–208

Page 548 new answer ——— 29 Aug 1995

32. (a) $U_n - Z_n/m_1 \equiv (m_2 - m_1)Y_n/m_1m_2$ (modulo 1), and $(m_1 - m_2)/m_1m_2 \approx 2^{-54}$. (Therefore we can analyze the high-order bits of Z_n by analyzing U_n. The low-order bits are probably random too, but this argument does not apply to them.) (b) We have $U_n = W_n/m$ for all n. The Chinese remainder theorem tells us that we need only verify the congruences $W_n \equiv X_n m_2$ (modulo m_1) and $W_n \equiv -Y_n m_1$ (modulo m_2), because $m_1 \perp m_2$. [Pierre L'Ecuyer and Shu Tezuka, *Math. Comp.* **57** (1991), 735–746.]

▶**Page 549** lines 1 and 2 of answer 3 ——— 28 Oct 1995

will be sufficiently ... But if $\leadsto$ will deviate from the correct distribution by at most $1/m$, as in exercise 2; but all of the excess is given to the smallest results. Thus if $k \approx m/3$, the result will be less than $k/2$ about $2/3$ of the time. It is much better to obtain a perfectly uniform distribution by rejecting U if $U > k\lfloor m/k \rfloor$; see D. E. Knuth, *The Stanford GraphBase* (ACM Press, 1994), 221.

On the other hand, if

▶**Page 550** line 7 ——— 28 Jul 1981

q_k and $n = k$. Repeat the following operation $\leadsto$ q_k. Set $n \leftarrow k$; then repeat the following operations

▶**Page 550** line 10 ——— 06 May 1995

$p_j \le 1/k \leadsto p_j < 1/k$

▶**Page 550** line −2 of answer 10 ——— 04 Apr 1981

$b_1, \ldots, b_{15}) \leadsto (b_1, \ldots, b_{15})$

▶**Page 550** line 1 of answer 11 ——— 09 Apr 1981

$\int_0^x \leadsto \int_3^x$

▶**Page 552** line 1 of answer 21 ——— 18 Dec 1982

$1.71153 \leadsto 1.71553$

Page 553 line 5 ——— 20 Mar 1995

(1980), to appear $\leadsto$ **25** (1980), 193–208

Page 554 lines 2 and 3 ——— 04 May 1981

Software, to appear.] $\leadsto$ *Software* **6** (1980), 359–364. This technique was introduced in the 1960s by David Seneschol; cf. *Amer. Statistician* **26**, 4 (October 1972), 56–57. The alternative of generating n uniform numbers and sorting them is probably faster, with an appropriate sorting method, but the method suggested here is particularly valuable if only a few of the largest or smallest X's are desired. Notice that $\bigl(F^{-1}(X_1), \ldots, F^{-1}(X_n)\bigr)$ will be sorted deviates having distribution F.]

Page 554 new answer ——— 24 Sep 1981

30. Generate random numbers $Z_1 = -\mu^{-1} \ln U_1$, $Z_2 = Z_1 - \mu^{-1} \ln U_2$, $\ldots$, until $Z_{m+1} \ge 1$. Output $(X_j, Y_j) = f(Z_j)$ for $1 \le j \le m$, where $f\bigl((.b_1 b_2 \ldots b_{2r})_2\bigr) = \bigl((.b_1 b_2 \ldots b_r)_2, (.b_{r+1} b_{r+2} \ldots b_{2r})_2\bigr)$. If the less significant bits are significantly less random than the more significant bits, it's safer (but slower) to let $f\bigl((.b_1 b_2 \ldots b_{2r})_2\bigr) = \bigl((.b_1 b_3 \ldots b_{2r-1})_2, (.b_2 b_4 \ldots b_{2r})_2\bigr)$.

Page 554 new answer ——— 24 Aug 1995

31. Use U_1 as the seed for *another* random number generator (e.g., a linear congruential generator with a different multiplier); take U_2, U_3, $\ldots$ from that one.

Page 554 line 2 of answer 5 ——— 06 Jul 1982

$\sum_{0 \le k \le N} k q_k \leadsto \sum_{1 \le k \le N} k q_k$

Page 554 new answer ——— 22 Aug 1995

8. (a) $p(0,k) = \binom{N-k}{n}/\binom{N}{n} = \binom{N-n}{k}/\binom{N}{k}$ of the $\binom{N}{n}$ samples omit the first k records.

(b) Set $X \leftarrow k-1$, where k is minimum with $U \ge \Pr(X \ge k)$. Thus, start with $X \leftarrow 0$, $p \leftarrow N-n$, $q \leftarrow N$, $R \leftarrow p/q$, and while $U < R$ set $X \leftarrow X+1$, $p \leftarrow p-1$, $q \leftarrow q-1$, $R \leftarrow Rp/q$. (This method is good when n/N is, say, $\ge 1/5$. We can assume that $n/N \le 1/2$; otherwise it's better to select $N-n$ *unsampled* items.)

(c) $\Pr\bigl(\min(Y_N,\ldots,Y_{N-n+1}) \ge k\bigr) = \prod_{j=0}^{n-1}\Pr(Y_{N-j} \ge k) = \prod_{j=0}^{n-1}(N-j-k)/(N-j)$. (This method is good if, say, $n \le 5$.)

(d) (Cf. exercise 3.4.1–29.) The value $X \leftarrow \lfloor N(1-U^{1/n})\rfloor$ if $n = O(\sqrt{N})$, or $X \leftarrow \lfloor \ln U/\ln\bigl(1-(n-1)/(N-1)\bigr)\rfloor$ if $n = \Omega(\sqrt{N})$, needs to be rejected with probability only $O(n/N)$. Precise details are worked out carefully in *CACM* **27** (1984), 703–718, and *ACM Trans. Math. Software* **13** (1987), 58–67. (This method is good when, say, $5 < n < \frac{1}{5}N$.)

After skipping X records and selecting the next, we decrease n and N by 1 and repeat the process until $n = 0$. A similar approach speeds up the reservoir method; see *ACM Trans. Math. Software* **11** (1985), 37–57.

▸**Page 555** answer 15 ——— 06 May 1995

line −3: $\lceil nk/N\rceil \rightsquigarrow \lfloor 2n(k-1)/N\rfloor$
line −1: table. $\rightsquigarrow$ table. [See *CACM* **29** (1986), 366–367.]

▸**Page 555** answer 16 ——— 22 Feb 1983

Delete this answer.

Page 558 replacement for answer 21 ——— 22 Aug 1995

20. (a) 2 and 1/2. (When n increases, we break $l_n^{(1)}$ in half.)

(b) Each new point breaks a single interval into two parts. Let ρ be equal to $\max_{k=0}^{n-1}\bigl((n+k)l_{n+k}^{(1)}\bigr)$. Then $1 = \sum_{k=1}^{n} l_n^{(k)} \le \sum_{k=0}^{n-1} l_{n+k}^{(1)} \le \sum_{k=0}^{n-1}\rho/(n+k) = \rho\ln 2 + O(1/n)$. So infinitely many m have $ml_m^{(1)} \ge 1/\ln 2 + O(1/m)$.

(c) To verify the hint, let $l_{2n}^{(k)}$ come from the interval with endpoints U_m and $U_{m'}$, and set $a_k = \max(m-n, m'-n, 1)$. Then $\rho = \min_{m=n+1}^{2n} ml_m^{(m)}$ implies $1 = \sum_{k=1}^{2n} l_{2n}^{(k)} \ge \sum_{k=1}^{2n}\rho/(n+a_k) \ge 2\rho\sum_{k=1}^{n}1/(n+k)$; hence $2\rho \le 1/(H_{2n}-H_n) = 1/\ln 2 + O(1/n)$.

(d) We have $(l_n^{(1)},\ldots,l_n^{(n)}) = (\lg\frac{n+1}{n}, \lg\frac{n+2}{n+1},\ldots,\lg\frac{2n}{2n-1})$, because the $(n+1)$st point always breaks the largest interval into intervals of length $\lg\frac{2n+1}{2n}$ and $\lg\frac{2n+2}{2n+1}$. [*Indagationes Math.* **11** (1949), 14–17.]

21. (a) No! We have $\overline{\Pr}(W_n < \frac{1}{2}) \ge \limsup_{n\to\infty}\nu\bigl(\lceil 2^{n-1/2}\rceil\bigr)/\lceil 2^{n-1/2}\rceil = 2-\sqrt{2}$, and $\underline{\Pr}(W_n < \frac{1}{2}) \le \liminf_{n\to\infty}\nu(2^n)/2^n = \sqrt{2}-1$, because $\nu\bigl(\lceil 2^{n-1/2}\rceil\bigr) = \nu(2^n) = \frac{1}{2}\sum_{k=0}^{n}(2^{k+1/2}-2^k) + O(n)$.

(b), (c). See *Indagationes Math.* **40** (1978), 527–541.

▸**Page 558** line 5 of answer 22 ——— 06 May 1995

$$\sum_{|c_1|,\ldots,|c_k|>r} \rightsquigarrow \sum_{\max\{|c_1|,\ldots,|c_k|\}>r}$$

Page 558 replacements for answers 23 and 24 ——— 11 Dec 1993

23. (a) This follows immediately from exercise 22. (b) Use a discrete Fourier transform in an analogous way; cf. *AMM* **75** (1968), 260–264.

24. (a) Let c be any nonzero integer; we must show, by exercise 22, that

$$\frac{1}{N}\sum_{n=0}^{N-1} e^{2\pi i c U_n} \to 0 \qquad \text{as} \qquad N\to\infty.$$

This follows because, if K is any positive integer, we have $\sum_{k=0}^{K-1}\sum_{n=0}^{N-1} e^{2\pi i c U_{n+k}} = K\sum_{n=0}^{N-1} e^{2\pi i c U_n} + O(K^2)$; hence

$$\begin{aligned}\frac{1}{N^2}\biggl|\sum_{n=0}^{N-1} e^{2\pi i c U_n}\biggr|^2 &= \frac{1}{K^2N^2}\biggl|\sum_{n=0}^{N-1}\sum_{k=0}^{K-1} e^{2\pi i c U_{n+k}}\biggr|^2 + O\Bigl(\frac{K}{N}\Bigr)\\ &\le \frac{1}{K^2N}\sum_{n=0}^{N-1}\biggl|\sum_{k=0}^{K-1} e^{2\pi i c U_{n+k}}\biggr|^2 + O\Bigl(\frac{K}{N}\Bigr)\\ &= \frac{1}{K} + \frac{2}{K^2N}\Re\biggl(\sum_{0\le j<k<K}\sum_{n=0}^{N-1} e^{2\pi i c(U_{n+k}-U_{n+j})}\biggr) + O\Bigl(\frac{K}{N}\Bigr) \to \frac{1}{K},\end{aligned}$$

by Cauchy's inequality. (b) When $d=1$, exercise 22 tells us that $\langle(\alpha_1 n+\alpha_0) \bmod 1\rangle$ is equidistributed iff α_1 is irrational. When $d>1$, we can use (a) and induction on d. [*Acta Math.* **56** (1931), 373–456. The result in (b) had previously been obtained in a more complicated way by H. Weyl, *Nachr. Gesellschaft der Wiss. Göttingen*, Math.-Phys. Kl. (1914), 234–244. A similar argument proves that the polynomial sequence is equidistributed if at least one of the coefficients $\alpha_d, \ldots, \alpha_1$ is irrational.]

Page 561 line 3 of answer 37 ______ 20 Mar 1981

334.] $\leadsto$ 334; see also the Ph.D. thesis of Thomas N. Herzog, Univ. of Maryland (1975).]

Page 561 new answers ______ 08 Aug 1995

40. Since F_k depends only on $B_1\ldots B_k$, we have $P(A_k^P, \$_N) = \frac12$. Let $q(B_1\ldots B_k) = \Pr(B_{k+1}=1 \mid B_1\ldots B_k)$, where the probability is taken over all elements of S having $B_1\ldots B_k$ as the first k bits. Similarly, let $q_b(B_1\ldots B_k) = \Pr(F_k=1 \text{ and } B'_{k+1}=b \mid B_1\ldots B_k)$. Then we have $\Pr(A_k^P=1 \mid B_1\ldots B_k) = \Pr\bigl((F_k+B_{k+1}+B'_{k+1}) \bmod 2 = 1 \mid B_1\ldots B_k\bigr) = q\cdot(\frac12-q_0+q_1)+(1-q)\cdot(q_0+\frac12-q_1) = \frac12-(q_0+q_1)+2\bigl(qq_1+(1-q)q_0\bigr) = \frac12 - \Pr(F_k=1 \mid B_1\ldots B_k) + 2\Pr(F_k=1 \text{ and } B'_{k+1}=B_{k+1} \mid B_1\ldots B_k)$. Hence $\Pr(A_k^P=1) = \sum_{B_1\ldots B_k}\Pr(B_1\ldots B_k)\Pr(A_k^P=1 \mid B_1\ldots B_k) = \frac12 - \Pr(F_k=1) + \Pr(F_{k+1}=1)$. [Cf. Theorem 4 of Goldreich, Goldwasser, and Micali in *JACM* **33** (1986), 792–807.]

41. Choose k uniformly from $\{0,\ldots,N-1\}$ and use the construction in the proof of Lemma P1. Then the proof of P1 shows that A' will be equal to 1 with probability $\sum_{k=0}^{N-1}(\frac12-p_k+p_{k+1})/N$.

42. (a) Let $X = X_1+\cdots+X_n$. Clearly $\mathrm{E}\,X = n\mu$; and we have $\mathrm{E}\bigl((X-n\mu)^2\bigr) = \mathrm{E}\,X^2 - n^2\mu^2 = n\,\mathrm{E}\,X_j^2 + 2\sum_{1\le i<j\le n}(\mathrm{E}\,X_i)(\mathrm{E}\,X_j) - n^2\mu^2 = n\,\mathrm{E}\,X_j^2 - n\mu^2 = n\sigma^2$. Also $\mathrm{E}\bigl((X-n\mu)^2\bigr) = \sum_{x\ge0} x\Pr\bigl((X-n\mu)^2=x\bigr) \ge \sum_{x\ge tn\sigma^2} x\Pr\bigl((X-n\mu)^2=x\bigr) \ge \sum_{x\ge tn\sigma^2} tn\sigma^2\Pr\bigl((X-n\mu)^2=x\bigr) = tn\sigma^2\Pr\bigl((X-n\mu)^2\ge tn\sigma^2\bigr)$.

(b) There is a position i where $c_i \ne c'_i$, say $c_i=0$ and $c'_i=1$. Then there's a position j where $c_j=1$. For any fixed setting of B in the $k-2$ rows other than i or j, we have $(cB, c'B) = (d, d')$ if and only if rows i and j have particular values; this occurs with probability $1/2^{2R}$.

(c) In the notation of Algorithm L, take $n = 2^k - 1$ and $X_c = (-1)^{G(cB+e_i)}$; then $\mu = s$ and $\sigma^2 = 1 - s^2$. The probability that $X = \sum_{c\neq 0} X_c$ is negative is at most the probability that $(X - n\mu)^2 \geq n^2\mu^2$. By (a) this is at most $\sigma^2/(n\mu^2)$.

43. The conclusion for fixed M would be of no interest, since there obviously exists an algorithm to factor any fixed M (namely, an algorithm that knows the factors). The theory applies to *all* algorithms that have short running time, not only to algorithms that are effectively discoverable.

[The answer previously numbered 40 now becomes number 44.]

Page 561 answer 3.6–7 ________________ 01 Aug 1995

"Deciphering a linear congruential encryption," ... problems.
IEEE Trans. **IT-31** (1985), 49–52. J. Reeds, *Cryptologia* **1** (1977), 20–26, **3** (1979), 83–95, initiated the study of related problems. See also L. Blum, M. Blum, and M. Shub, *SICOMP* **15** (1986), 364–383; J. Boyar, *J. Cryptology* **1** (1989), 177–184. In *SICOMP* **17** (1988), 262–280, Frieze, Håstad, Kannan, Lagarias, and Shamir discuss general techniques that are useful in problems like this.

Page 561 new answer ________________ 26 Aug 1995

8. The values of $X_0, \ldots, X_{99}$ are not all even. The polynomial $z^{100} + z^{37} + 1$ is primitive (see Section 3.2.2); hence there is a number $h(s)$ such that $P_0(z) \equiv z^{h(s)}$ (modulo 2 and $z^{100} + z^{37} + 1$). Now $zP_{n+1}(z) = P_n(z) - X_n z^{37} - X_{n+63} + X_{n+63}z^{100} + X_{n+100}z^{37} \equiv P_n(z) + X_{n+63}(z^{100} + z^{37} + 1)$ (modulo 2), so the result follows by induction on n.

(b) The operations "square" and "multiply by z" in *ran_start* change $p(z) = x_{99}z^{99} + \cdots + x_0$ to $p(z)^2$ and $zp(z)$, respectively, modulo 2 and $z^{100} + z^{37} + 1$, because $p(z)^2 \equiv p(z^2)$. (We consider here only the low-order bits. The other bits are manipulated in an ad hoc way that tends to preserve and/or enhance whatever disorder they already have.) Therefore if $s = (1s_j \ldots s_0)_2$ we have $h(s) = (1s_0 \ldots s_j 1)_2 \cdot 2^{69}$.

(c) $z^{h(s)-n} \equiv z^{h(s')-n'}$ (modulo 2 and $z^{100} + z^{37} + 1$) implies that $h(s) - n \equiv h(s') - n'$ (modulo $2^{100} - 1$). Since $2^{69} \leq h(s) < 2^{100} - 2^{69}$, we have $|n - n'| \geq |h(s) - h(s')| \geq 2^{70}$.

[This method of initialization was inspired by comments of R. P. Brent, *Proc. Australian Supercomputer Conf.* **5** (1992), 95–104, although Brent's algorithm was completely different. In general if the lags are $k > l$, if $0 \leq s < 2^e$, and if the separation parameter satisfies $t + e \leq k$, this method of proof shows that $|n - n'| \geq 2^t - 1$, with $2^t - 1$ occurring only if $\{s, s'\} = \{0, 2^e - 1\}$.]

Page 561 new answers ________________ 27 Aug 1995

9. The following code belongs to the simplified language Subset FORTRAN, as defined by the American National Standards Institute, except for its use of `PARAMETER` statements for readability.

```
      SUBROUTINE RNARRY(AA,N)
      IMPLICIT INTEGER (A-Z)
      DIMENSION AA(*)
      PARAMETER (KK=100)
      PARAMETER (LL=37)
```

```
      PARAMETER (MM=2**30)
      COMMON /RSTATE/ RANX(KK)
      SAVE /RSTATE/
      DO 1 J=1,KK
 1       AA(J)=RANX(J)
      DO 2 J=KK+1,N
         AA(J)=AA(J-KK)-AA(J-LL)
         IF (AA(J) .LT. 0) AA(J)=AA(J)+MM
 2    CONTINUE
      DO 3 J=1,LL
         RANX(J)=AA(N+J-KK)-AA(N+J-LL)
         IF (RANX(J) .LT. 0) RANX(J)=RANX(J)+MM
 3    CONTINUE
      DO 4 J=LL+1,KK
         RANX(J)=AA(N+J-KK)-RANX(J-LL)
         IF (RANX(J) .LT. 0) RANX(J)=RANX(J)+MM
 4    CONTINUE
      END

      SUBROUTINE RNSTRT(SEED)
      IMPLICIT INTEGER (A-Z)
      PARAMETER (KK=100)
      PARAMETER (LL=37)
      PARAMETER (MM=2**30)
      PARAMETER (TT=70)
      PARAMETER (KKK=KK+KK-1)
      DIMENSION X(KKK)
      COMMON /RSTATE/ RANX(KK)
      SAVE /RSTATE/
      SS=SEED-MOD(SEED,2)+2
      DO 1 J=1,KK
         X(J)=SS
         SS=SS+SS
         IF (SS .GE. MM) SS=SS-MM+2
 1    CONTINUE
      DO 2 J=KK+1,KKK
 2       X(J)=0
      X(2)=X(2)+1
      SS=SEED
      T=TT-1
 10   DO 12 J=KK,2,-1
 12      X(J+J-1)=X(J)
      DO 13 J=KKK,KK-LL+1,-2
 13      X(KKK-J+2)=X(J)-MOD(X(J),2)
      DO 14 J=KKK,KK+1,-1
         IF (MOD(X(J),2) .EQ. 1) THEN
            X(J-(KK-LL))=X(J-(KK-LL))-X(J)
            IF (X(J-(KK-LL)) .LT. 0) X(J-(KK-LL))=X(J-(KK-LL))+MM
            X(J-KK)=X(J-KK)-X(J)
```

```
            IF (X(J-KK) .LT. 0) X(J-KK)=X(J-KK)+MM
         END IF
 14   CONTINUE
      IF (MOD(SS,2) .EQ. 1) THEN
         DO 16 J=KK,1,-1
 16      X(J+1)=X(J)
         X(1)=X(KK+1)
         IF (MOD(X(KK+1),2) .EQ. 1) THEN
            X(LL+1)=X(LL+1)-X(KK+1)
            IF (X(LL+1) .LT. 0) X(LL+1)=X(LL+1)+MM
         END IF
       END IF
       IF (SS .NE. 0) THEN
          SS=SS/2
       ELSE
          T=T-1
       END IF
       IF (T .GT. 0) GO TO 10
       DO 20 J=1,LL
 20       RANX(J+KK-LL)=X(J)
       DO 21 J=LL+1,KK
 21       RANX(J-LL)=X(J)
       END
```

10. Floating-point arithmetic on 64-bit operands conforming to ANSI/IEEE Standard 754 allows us to compute $U_n = (U_{n-100} - U_{n-37}) \bmod 1$ with perfect accuracy for fractions U_n that are integer multiples of 2^{-53}. However, the following program uses the *additive* recurrence $U_n = (U_{n-100} + U_{n-37}) \bmod 1$ on integer multiples of 2^{-52} instead, because pipelined computers can subtract an integer part more quickly than they can branch conditionally on the sign of an intermediate result. The theory of exercise 8 applies equally well to this sequence. The main new idea in *ranf_start* is to maintain a copy in *ul* of the least-significant bits of the fractions in *u*. A FORTRAN translation similar to the code in exercise 9 will generate exactly the same numbers as this C routine.

```
#define KK 100                                    /* the long lag */
#define LL  37                                   /* the short lag */
#define mod_sum(x,y) (((x)+(y))-(int)((x)+(y)))   /* (x+y) mod 1.0 */
double ran_u[KK];                           /* the generator state */
void ranf_array(double aa[],int n) { /* aa gets n random fractions */
  register int i,j;
  for (j=0;j<KK;j++) aa[j]=ran_u[j];
  for (;j<n;j++) aa[j]=mod_sum(aa[j-KK],aa[j-LL]);
  for (i=0;i<LL;i++,j++) ran_u[i]=mod_sum(aa[j-KK],aa[j-LL]);
  for (;i<KK;i++,j++) ran_u[i]=mod_sum(aa[j-KK],ran_u[i-LL]);
}
#define TT  70              /* guaranteed separation between streams */
#define is_odd(s) ((s)&1)
void ranf_start(long seed) {     /* do this before using ranf_array */
```

```
  register int t,s,j;
  double u[KK+KK-1],ul[KK+KK-1];
  double ulp=(1.0/(1<<30))/(1<<22);                /* 2 to the -52 */
  double ss=2.0*ulp*(seed+2);
  for (j=0;j<KK;j++) {
    u[j]=ss; ul[j]=0.0;                     /* bootstrap the buffer */
    ss+=ss; if (ss>=1.0) ss-=1.0-2*ulp;  /* cyclic shift of 51 bits */
  }
  for (;j<KK+KK-1;j++) u[j]=ul[j]=0.0;
  u[1]+=ulp;ul[1]=ulp;            /* make u[1] (and only u[1]) "odd" */
  s=seed;
  t=TT-1; while (t) {
    for (j=KK-1;j>0;j--) ul[j+j]=ul[j],u[j+j]=u[j];     /* "square" */
    for (j=KK+KK-2;j>KK-LL;j-=2)
        ul[KK+KK-1-j]=0.0,u[KK+KK-1-j]=u[j]-ul[j];
    for (j=KK+KK-2;j>=KK;j--) if(ul[j]) {
      ul[j-(KK-LL)]=ulp-ul[j-(KK-LL)],
        u[j-(KK-LL)]=mod_sum(u[j-(KK-LL)],u[j]);
      ul[j-KK]=ulp-ul[j-KK],u[j-KK]=mod_sum(u[j-KK],u[j]);
    }
    if (is_odd(s)) {                            /* "multiply by z" */
      for (j=KK;j>0;j--)  ul[j]=ul[j-1],u[j]=u[j-1];
      ul[0]=ul[KK],u[0]=u[KK];       /* shift the buffer cyclically */
      if (ul[KK]) ul[LL]=ulp-ul[LL],u[LL]=mod_sum(u[LL],u[KK]);
    }
    if (s) s>>=1; else t--;
  }
  for (j=0;j<LL;j++) ran_u[j+KK-LL]=u[j];
  for (;j<KK;j++) ran_u[j-LL]=u[j];
}
```

Page 561 another new answer ________________ 30 Aug 1995

11. A simple linear congruential generator like (1) would fail, because m would be much too small. Good results are possible by combining three (not two) such generators, with multipliers and moduli $(157, 32363)$, $(146, 31727)$, $(142, 31657)$, as suggested by P. L'Ecuyer in *CACM* **31** (1988), 747–748. However, the best method is probably to use the C programs *ran_array* and *ran_start*, with the following changes to keep all numbers in range: '`long`' becomes '`int`'; '`MM`' is defined to be '`(1U<<15)`'; and the type of variable *ss* should be **unsigned int**. This generates 15-bit integers, all of whose bits are usable. The seed is now restricted to the range $[0\,.\,.\,32765]$.

Page 561 new answer ________________ 19 Sep 1995

12. A program for subtract-with-borrow would be very similar to *ran_array*, but slower because of the carry maintenance. As in exercise 10, floating-point arithmetic could be used with perfect accuracy. It is possible to guarantee disjointness of the sequences produced from different seeds s by initializing the generator with the $(-n)$th element of the sequence, where $n = 2^{70s}$; this requires computing $b^n \bmod (b^k - b^l \pm 1)$. Squaring a radix-$b$ number mod $b^k - b^l \pm 1$ is, however, considerably more complicated than the analogous operation in program *ran_start*, and for k in a practical range it takes about $k^{1.6}$ operations instead of $O(k)$.

Both methods generate sequences of the same quality in practice, when they have roughly the same value of k. The only significant difference between them is a better theoretical guarantee and a provably immense period for the subtract-with-borrow method; the analysis of lagged Fibonacci generators is less complete. Experience shows that we should not reduce the value of k in subtract-with-borrow just because of these theoretical advantages. When all is said and done, lagged Fibonacci generators seem preferable from a practical standpoint; the subtract-with-borrow method is then valuable chiefly because of the insight it gives us into the excellent behavior of the simpler approach.

▸**Page 563** answer 16 ———————— 07 Jul 1983

line 5: $\alpha^p \leadsto \alpha^P$
line 6: $(\alpha^p)_{i-1} \leadsto (\alpha^P)_{i-1}$
line 8: $(\alpha 0)^p \leadsto (\alpha 0)^P$ $\quad (\alpha x 1)^p \leadsto (\alpha x 1)^P$ $\quad \alpha^p 1 \leadsto \alpha^P 1$

Page 564 new paragraph to precede answer 19 ———————— 06 May 1995

I. Kátai and J. Szabó have shown that the radix $-d+i$ yields a number system with digits $\{0, 1, \ldots, d^2\}$ [*Acta Scient. Math.* **37** (1975), 255–260. Further properties of such systems have been investigated by W. J. Gilbert, *Canadian J. Math.* **34** (1982), 1335–1348; *Math. Magazine* **57** (1984), 77–81. For studies of number systems based on more general algebraic integers, see I. Kátai and B. Kovács, *Acta Math. Acad. Sci. Hung.* **37** (1981), 159–164, 405–407; B. Kovács, *Acta Math. Hung.* **58** (1991), 113–120; B. Kovács and A. Pethő, *Studia Scient. Math. Hung.* **27** (1992), 169–172.

Page 564 answer 19 ———————— 31 Dec 1982

line 9: Reference: $\leadsto$ References:
line 10: 1–9. $\leadsto$ 1–9; *JACM* **29** (1982), 1131–1143.

Page 565 answer 23 ———————— 21 Mar 1981

line 4: zero since it is $\leadsto$ zero if $0 \in D$, since T is
line 5: $10^k \leadsto b^k$
line 6; zero. $\leadsto$ zero. On the other hand, as pointed out by K. A. Brakke, every real number has infinitely many representations in the number system of exercise 21.
line 9: less $\leadsto$ fewer

▸**Page 565** line 4 of answer 24 ———————— 20 Nov 1981

superflous $\leadsto$ superfluous

Page 567 answer 32 ———————— 06 May 1995

generalization $\leadsto$ constructively proved generalization

▸**Page 567** and 568, answer 33 ———————— 18 Mar 1995

page 567, lines 7 and 9: $S_i \leadsto s_i$
page 568, line 11: appears $\leadsto$ appears among
page 568, line 14 should read as follows:
$[|S| \le 1] + z \sum_{T \in \mathcal{T}(S)} c_T k_T(z)$. [Cf. *J. Algorithms* **2** (1981), 31–43.]
page 568, line 19: $l_{012}(z) \leadsto k_{012}(z)$

Page 568 new answer ——— 24 Aug 1996

34. There is exactly one string α_n on the symbols $\{\bar{1}, 0, 1\}$ such that $n = (\alpha_n)_2$ and α_n has no leading zeros or consecutive nonzeros: α_0 is empty, otherwise $\alpha_{2n} = \alpha_n 0$, $\alpha_{4n+1} = \alpha_n 01$, $\alpha_{4n-1} = \alpha_n 0\bar{1}$. Any string that represents n can be converted to α_n by using the reductions $1\bar{1} \to 01$, $\bar{1}1 \to 0\bar{1}$, $01\ldots11 \to 10\ldots0\bar{1}$, $0\bar{1}\ldots\bar{1}\bar{1} \to \bar{1}0\ldots01$, and inserting or deleting leading zeros. Since these reductions do not increase the number of nonzero digits, α_n has the fewest. [*Advances in Computers* **1** (1960), 244–260.] The number of nonzero digits in α_n, denoted by $\overline{\nu}(n)$, is the number of 1's in the ordinary representation that are immediately preceded by 0 or by the substring $00(10)^k 1$ for some $k \ge 0$.

A generalization to radix $b > 2$ has been given by J. von zur Gathen, *Computational Complexity* **1** (1991), 360–394.

Page 568 replacement for answer 4.2.1–1 ——— 10 Jan 1981

1. $N = (62, +.60\,22\,52\,00)$; $h = (37, +.66\,25\,60\,00)$. Note that the quantity $10h$ would be $(38, +.06\,62\,56\,00)$.

▸**Page 570** after line 9 ——— 15 Jan 1981

insert a new instruction '`JXNZ *+3`' between '`ENT2 0`' and '`JAZ *+2`'

▸**Page 570** line 2 of answer 19 ——— 15 Jan 1981

$b/20 \rightsquigarrow b/2\,0$

▸**Page 571** line 3 of answer 11 ——— 22 May 1997

$|t| < 2b^e \rightsquigarrow |t| < b^e$

Page 573 line 5 ——— 09 Jul 1996

See also $\rightsquigarrow$ When some x's are negative and others are positive, we can combine them advantageously; see T. O. Espelid, *SIAM Review* **37** (1995), 603–607. See also

▸**Page 573** replacement for answer 23 ——— 02 Feb 1981

23. If $u \ge 0$ or $u \le -1$ we have $u \,(\mathrm{mod})\, 1 = u \bmod 1$, so the identity holds. If $-1 < u < 0$, then $u \,(\mathrm{mod})\, 1 = u \oplus 1 = u + 1 + r$ where $|r| \le \frac{1}{2}b^{-p}$; the identity holds iff $\mathrm{round}(1 + r) = 1$, so it always holds if we round to even. With the text's rounding rule the identity fails iff b is a multiple of 4 and $-1 < u < 0$ and $u \bmod 2b^{-p} = \frac{3}{2}b^{-p}$ (e.g., $p = 3$, $b = 8$, $u = -(.0124)_8$).

Page 574 new answer ——— 22 Aug 1996

28. According to the theory in Section 4.5.3, the convergents to the continued fraction $\sqrt{3} = 1 + /\!/1, 2, 1, 2, \ldots /\!/$ are $p_n/q_n = K_{n+1}(1, 1, 2, 1, 2, \ldots)/K_n(1, 2, 1, 2, \ldots)$. These convergents are excellent approximations to $\sqrt{3}$, hence $3q_n^2 \approx p_n^2$; in fact, $3q_n^2 - p_n^2 = 2 - 3(n \bmod 2)$. The example given is $2p_{31}^2 + (3q_{31}^2 - p_{31}^2)(3q_{31}^2 + p_{31}^2) = 2p_{31}^2 - (p_{31}^2 - 1 + p_{31}^2) = 1$. Floating point subtraction of p_{31}^2 from $3q_{31}^2$ yields zero, unless we can represent $3q_{31}^2$ almost perfectly; subtracting p_{31}^4 from $9q_{31}^4$ generally gives rounding errors much larger than $2p_{31}^2$. Similar examples can be based on continued fraction approximations to any algebraic number.

Page 574 new answer ——— 17 Apr 1995

29. Let $e = \max(e_u, e_{u'})$, $e' = \max(e_v, e_{v'})$, $e'' = \max(e_{u\oplus v}, e_{u'\oplus v'})$, and assume that $q = 0$. Then $(u \oplus v) - (u' \oplus v') \le u + v + \frac{1}{2}b^{e''-p} - u' - v' + \frac{1}{2}b^{e''-p} \le \epsilon b^e + \epsilon b^{e'} + b^{e''-p}$, and $e'' \ge \max(e, e')$. Hence $u \oplus v \sim u' \oplus v'\ (2\epsilon + b^{-p})$.

If $b = 2$ this estimate can be improved to $1.5\epsilon + b^{-p}$. For $\epsilon + b^{-p}$ is an upper bound if $u - u'$ and $v - v'$ have opposite signs, and in the other case we cannot have $e = e' = e''$.

Page 578 line 1 of answer 17 ——— 27 Sep 1995

Fibonacci Quarterly **7** (1969), 474–475. (Persi $\leadsto$ *J. Math. Soc. Japan* **4** (1952), 313–322. (The fact that harmonic probability extends ordinary probability follows from a theorem of Cesàro, [*Atti della Reale Accademia dei Lincei, Rendiconti* (4) **4** (1888), 452–457]. Persi

▶**Page 578** line 6 of answer 17 ——— 27 Sep 1995

logarithmic probability $\leadsto$ harmonic probability

Page 579 new answer ——— 27 Sep 1995

19. Equivalently, $\langle(\log 10F_n) \bmod 1\rangle$ is equidistributed in the sense of Definition 3.5B. Since $\log_{10} F_n = n\log 10\phi - \log_{10}\sqrt{5} + O(\phi^{-2n})$ by 1.2.8–(14), this is equivalent to equidistribution of $\langle n\log_{10}\phi\rangle$, which follows from ex. 3.5–22. [*Fibonacci Quarterly* **5** (1967), 137–140.] The same proof shows that the sequences $\langle b^n\rangle$ obey the logarithmic law for all integers $b > 1$ that aren't powers of 10 [Yaglom and Yaglom, *Challenging Problems with Elementary Solutions* (Moscow: 1954; English translation, 1964), Problem 91b].

Notes: Many other sequences of integers have this property. For example, P. Diaconis [*Annals of Probability* **5** (1977), 72–81] showed that $\langle n!\rangle$ is one such sequence, and that binomial coefficients obey the logarithmic law too, in the sense that

$$\lim_{n\to\infty}\frac{1}{n+1}\sum_{k=0}^{n}[10f_{\binom{n}{k}} < r] = \log_{10} r\,.$$

P. Schatte [*Math. Nachrichten* **148** (1990), 137–144] proved that the denominators of continued fraction approximations have logarithmic fraction parts whenever the partial quotients have a repeating pattern with polynomial variation as in exercise 4.5.3–16. One interesting open question is whether the sequence $\langle 2!, (2!)!, ((2!)!)!, \ldots\rangle$ has logarithmic fraction parts; see J. H. Conway and M. J. T. Guy, *Eureka* **25** (1962), 18–19.

▶**Page 581** lines 6 and 7 ——— 26 May 1997

`INC2` $\leadsto$ `INC1`
`J2N` $\leadsto$ `J1N`

▶**Page 581** the first instruction in answer 13 ——— 26 Aug 1986

`ENTX N` $\leadsto$ `ENT1 N`

Page 582 new paragraph to follow answer 15 ——— 04 Oct 1995

[It follows that correctly rounded results of multiprecision floating point fractions can almost always be obtained by doing only about half the work needed to compute the full double-length product; moreover, a simple test will identify the rare cases for which full precision is needed. See W. Krandick and J. R. Johnson, *Proc. IEEE Symp. Computer Arithmetic* **11** (1993), 228–233.]

▶**Page 582** line 3 of answer 21 ——— 31 Dec 1989

$0 \le \alpha = q - u/v \leadsto 0 < \alpha = 1 + q - u/v$

Page 584 new answer ______ 07 Jan 1984

38. Set $k \leftarrow n$, $r \leftarrow 0$, $s \leftarrow 1$, $t \leftarrow 0$, $w \leftarrow u$; we will preserve the invariant relation $uv = 2^{2k}(r + s^2 - s) + 2^{2k-n}t + 2^{2k-2n}vw$ with $0 \le t, w < 2^n$, and with $0 < r \le 2s$ unless $(r, s) = (0, 1)$. While $k > 0$, let $4w = 2^n w' + w''$ and $4t + w'v = 2^n t' + t''$, where $0 \le w'', t'' < 2^n$ and $0 \le t' \le 6$; then set $t \leftarrow t''$, $w \leftarrow w''$, $s \leftarrow 2s$, $r \leftarrow 4r + t' - s$, $k \leftarrow k - 1$. If $r \le 0$, set $s \leftarrow s - 1$ and $r \leftarrow r + 2s$; otherwise, if $r > 2s$, set $r \leftarrow r - 2s$ and $s \leftarrow s + 1$ (this correction might need to be done twice). Repeat until $k = 0$. Then $uv = r + s^2 - s$, since w is always a multiple of 2^{2n-2k}. Consequently $r = 0$ iff $uv = 0$; otherwise the answer is s, because $uv - s \le s^2 < uv + s$.

Page 584 new answer ______ 19 Apr 1995

39. Given ϕ with an accuracy of $\pm 2^{-2n}$, we can successively compute ϕ^{-1}, ϕ^{-2}, ... by subtraction until $\phi^{-k} < 2^{-n}$; the accumulated error will not exceed 2^{1-n}. Then we can use the series $\ln\phi = \ln\bigl((1 + \phi^{-3})/(1 - \phi^{-3})\bigr) = 2(\phi^{-3} + \frac{1}{3}\phi^{-9} + \frac{1}{5}\phi^{-15} + \cdots)$. [Cf. William Schooling's article in *Napier Tercentenary Memorial*, ed. by C. G. Knott (London: Longmans, 1915), 337–344.] An even better procedure, suggested in 1965 by J. W. Wrench, Jr., is to evaluate $\ln\phi = \frac{1}{2}\ln\bigl((1+5^{-1/2})/(1-5^{-1/2})\bigr) = (2\phi - 1)(5^{-1} + \frac{1}{3}5^{-2} + \frac{1}{5}5^{-3} + \cdots)$.

Page 584 new answer ______ 17 Jan 1996

40. Let $S_j = \sum_{k\ge 0} 16^{-k}/(8k+j)$. We want to know whether or not $2^{n-1}\pi \bmod 1 < \frac{1}{2}$. Since $\pi = 4S_1 - 2S_4 - S_5 - S_6$, it suffices to have good estimates of $2^{n-1}S_j \bmod 1$. Now $2^{n-1}S_j$ is congruent (modulo 1) to $\sum_{0\le k<n/4} a_{njk}/(8k+j) + \sum_{k\ge n/4} 2^{n-1-4k}/(8k+j)$, where $a_{njk} = 2^{n-1-4k} \bmod (8k + j)$. Each term in the first sum can be approximated within 2^{-m} by computing a_{njk} in $O(\log n)$ operations (Section 4.6.3) and then finding the scaled quotient $\lfloor 2^m a_{njk}/(8k + j)\rfloor$. The second sum can be approximated within 2^{-m} by computing 2^m times its first $m/4$ terms. If $m \approx 2\lg n$, the range of uncertainty will be $\approx 1/n$, and this will almost always be accurate enough. [*Math. Comp.* **66** (1997), 903–913.]

Notes: Let $\zeta = e^{\pi i/4} = (1 + i)/\sqrt{2}$ be an 8th root of unity, and consider the values $l_j = \ln(1 - \zeta^j/\sqrt{2})$. Then $l_0 = \ln(1 - 1/\sqrt{2})$, $l_1 = \bar l_7 = \frac{1}{2}\ln\frac{1}{2} - i\arctan 1$, $l_2 = \bar l_6 = \frac{1}{2}\ln\frac{3}{2} - i\arctan(1/\sqrt{2})$, $l_3 = \bar l_5 = \frac{1}{2}\ln\frac{5}{2} - i\arctan(1/3)$, $l_4 = \ln(1 + 1/\sqrt{2})$. Also $-S_j/2^{j/2} = \frac{1}{8}(l_0 + \zeta^{-j}l_1 + \cdots + \zeta^{-7j}l_7)$ for $1 \le j \le 8$ by 1.2.9–(13). Therefore $4S_1 - 2S_4 - S_5 - S_6 = 2l_0 - (2-2i)2l_1 + 2l_4 + (2+2i)l_7 = \pi$. Other identities of interest are:

$$\begin{aligned}
\ln 2 &= S_2 + \tfrac{1}{2}S_4 + \tfrac{1}{4}S_6 + \tfrac{1}{8}S_8;\\
\ln 3 &= 2S_2 + \tfrac{1}{2}S_6;\\
\ln 5 &= 2S_2 + 2S_4 + \tfrac{1}{2}S_6;\\
\sqrt{2}\,\ln(\sqrt{2}+1) &= S_1 + \tfrac{1}{2}S_3 + \tfrac{1}{4}S_5 + \tfrac{1}{8}S_7;\\
\sqrt{2}\,\arctan(1/\sqrt{2}) &= S_1 - \tfrac{1}{2}S_3 + \tfrac{1}{4}S_5 - \tfrac{1}{8}S_7;\\
\arctan(1/3) &= S_1 - S_2 - \tfrac{1}{2}S_4 - \tfrac{1}{4}S_5;\\
0 &= 8S_1 - 8S_2 - 4S_3 - 8S_4 - 2S_5 - 2S_6 + S_7.
\end{aligned}$$

In general we have

$$\sum_{k\ge0}\frac{z^{8k+1}}{8k+1}=A+B+C+D,\qquad \sum_{k\ge0}\frac{z^{8k+5}}{8k+5}=A-B+C-D,$$
$$\sum_{k\ge0}\frac{z^{8k+3}}{8k+3}=A-B-C+D,\qquad \sum_{k\ge0}\frac{z^{8k+7}}{8k+7}=A+B-C-D,$$

where

$$A=\frac18\ln\frac{1+z}{1-z},\qquad B=\frac{1}{2^{7/2}}\ln\frac{1+\sqrt2z+z^2}{1-\sqrt2z+z^2},$$
$$C=\frac14\arctan z,\qquad D=\frac{1}{2^{5/2}}\arctan\frac{\sqrt2z}{1-z^2};$$

and

$$\sum_{k\ge0}\frac{z^{mk+a}}{mk+a}=-\frac1m\Bigl(\ln(1-z)+(-1)^a[m\,\mathrm{even}]\ln(1+z)+f_{am}(z)\Bigr),$$

$$f_{am}(z)=\sum_{k=1}^{\lfloor(m-1)/2\rfloor}\biggl(\cos\frac{2\pi ka}{m}\ln\Bigl(1-2z\cos\frac{2\pi k}{m}+z^2\Bigr)-2\sin\frac{2\pi ka}{m}\arctan\frac{z\sin(2\pi k/m)}{1-z\cos(2\pi k/m)}\biggr).$$

Page 584 new answer ______ 25 Aug 1996

41. To get the most significant $n/2$ places, we need about $\sum_{k=1}^{n/2}\approx\frac18n^2$ basic operations (cf. exercise 15). And we can get the least significant $n/2$ places by using a b-adic method when b is a power of 2 (cf. exercise 4.1–31): The problem is easily reduced to the case where v is odd. Let $u=(\ldots u_2u_1u_0)_b$, $v=(\ldots v_2v_1v_0)_b$, and $w=(\ldots w_2w_1w_0)_b$, where we want to solve $u=vw$ (modulo $b^{n/2}$). Compute v' such that $v'v \bmod b=1$ (see Section 4.5.2). Then $w_0=v'u_0 \bmod b$, and we can compute $u'=u-w_0v$, $w_1=v'u'_0 \bmod b$, etc. The rightmost $n/2$ places are found after about $\frac18n^2$ basic operations. So the total is $\frac14n^2+O(n)$, while Algorithm D needs about $n^2+O(n)$. A pure right-to-left method for all n digits would require $\frac12n^2+O(n)$. [See T. Jebelean, *J. Symbolic Comp.* **15** (1993), 169–180; A. Schönhage and E. Vetter, *Lecture Notes in Comp. Sci.* **855** (1994), 448–459.]

Page 584 new answer ______ 28 Jun 1995

42. (a) If $m=0$, let $v=u$. Otherwise subtract xw from $(u_{m+n-1}\ldots u_1u_0)_b$, where $x=u_0w' \bmod b$; this zeroes out the units digit, so we have effectively reduced m by 1. (This operation is closely related to the computation of u/w in b-adic arithmetic, since $u/w=q+b^mv/w$ for some integer q; cf. exercise 4.1–31. It wins over ordinary division because we never have to correct a trial divisor.)

(b) Apply (a) to the product uv. Memory space is conserved if we interlace multiplication and modulation as follows: Set $k\leftarrow0$, $t\leftarrow0$. Then while $k<n$, preserve the invariant relation $b^kt\equiv(u_{k-1}\ldots u_0)v$ (modulo w) by setting $t\leftarrow t+u_kv$, $t\leftarrow(t-xw)/b$, $k\leftarrow k+1$, where $x=t_0w' \bmod b$ is chosen to make $t-xw$ a multiple of b. This solution assumes that t, u, and v have a signed-magnitude representation; we can work also with nonnegative numbers $<2w$ or with complement notations, as discussed by Shand and Vuillemin and by Kornerup, [*IEEE Symp. Computer Arithmetic* **11** (1993), 252–259, 277–283]. If n is large, the techniques of Section 4.3.3 speed up the multiplication.

(c) Represent all numbers congruent to u (modulo w) by an internal value $r(u)$ where $r(u) \equiv b^n u$. Then addition and subtraction are handled as usual, while multiplication is $r(uv) = \mathrm{bmult}\bigl(r(u), r(v)\bigr)$, where bmult is the operation of (b). At the beginning of the computation, replace each operand u by $r(u) = \mathrm{bmult}(u, a)$, using the precomputed constant $a = b^{2n} \bmod w$. At the end, replace each $r(u)$ by $u = \mathrm{bmult}\bigl(r(u), 1\bigr)$. [In the application to RSA encryption, we could redefine the coding scheme so that precomputation and postcomputation are unnecessary.]

Page 584 new answer ——— 12 Apr 1996

43. By exercise 1.2.4–35 we have $w = \lfloor W/2^{16} \rfloor$, where $W = (2^8{+}1)t = (2^8{+}1)(uv{+}2^7)$. Therefore if $xy/255 > c{+}\frac{1}{2}$, we have $c < 2^8$, hence $w \ge \lfloor (2^{16}(c{+}1){+}2^8{-}c)/2^{16} \rfloor \ge c{+}1$; if $xy/255 < c + \frac{1}{2}$, we have $w \le \lfloor (2^{16}(c+1) - c - 1)/2^{16} \rfloor = c$. [Cf. J. F. Blinn, *IEEE Computer Graphics and Applic.* **14**, 6 (November 1994), 78–82.]

Page 586 replacement for the first paragraph of answer 7 ——— 02 Oct 1995

7. We have $v_j m_{j-1} \ldots m_1 \equiv u_j - (v_{j-1} m_{j-2} \ldots m_1 + \cdots + v_1)$ and $C_j m_{j-1} \ldots m_1 \equiv 1$ (modulo m_j) by (22), (24), and (25); cf. P. A. Pritchard, *CACM* **27** (1984), 57.

Page 587 new answer ——— 01 Jan 1997

14. Find the cyclic convolution $(z_0, z_1, \ldots, z_{n-1})$ of floating-point approximations to $(a_0u_0, a_1u_1, \ldots, a_{n-1}u_{n-1})$ and $(a_0v_0, a_1v_1, \ldots, a_{n-1}v_{n-1})$, where the constants $a_k = 2^{-(kq \bmod n)/n}$ have been precomputed. The identities $u = \sum_{k=0}^{n-1} u_k a_k 2^{kq/n}$ and $v = \sum_{k=0}^{n-1} v_k a_k 2^{kq/n}$ now imply that $w = \sum_{k=0}^{n-1} t_k a_k 2^{kq/n}$ where $t_k \approx z_k/a_k$. If sufficient accuracy has been maintained, each t_k will be very close to an integer. The representation of w can readily be found from those integers. [R. Crandall and B. Fagin, *Math. Comp.* **62** (1994), 305–324.]

▸**Page 588** line 1 of answer 7 ——— 24 Jan 1983

$p_{k-1} < N \le p_k \leadsto p_{k-1} < n \le p_k$

▸**Page 588** line 3 of answer 10 ——— 30 May 1997

$S(p,q) = \leadsto |S(p,q)| =$ (twice)

Page 589 line 7 ——— 11 Nov 1980

SIAM J. Computing, to appear $\leadsto$ *SICOMP* **9** (1980), 490–508.

▸**Page 589** line 1 of answer 1 ——— 07 Sep 1989

$(\ldots(a_m b_{m-1} + a_{m-2})b_{m-2} + \cdots + a_1)b_1 + a_0 \leadsto (\ldots(a_m b_{m-1} + a_{m-1})b_{m-2} + \cdots + a_1)b_0 + a_0$

Page 589 new answer ——— 16 Sep 1995

16. If $n = 1$, just form the 2-place answer. If n is even, proceed recursively as follows, when $B = b^{n/2}$: Given $({*}{*}{*}{*}{*}{*}U_1U_0)_B$ and $({*}{*}{*}{*}{*}{*}V_1V_0)_B$, move the data to $({*}{*}{*}U_1{*}U'{*}V_0)_B$ and $({*}{*}{*}V_1{*}V'{*}V_0)_B$ where $U' = |U_1{-}U_0|$, $V' = |V_1{-}V_0|$; then use the method for $n/2$ to form $(w_{21}w_{20}w_{11}w_{10}w_{01}w_{00}{*}{*})_B$ in the u array, where $W_2 = U_1V_1$, $W_1 = U'V'$, $W_0 = U_0V_0$. Using the V array for intermediate storage, change the U array to the final answer $X = (x_3x_2x_1x_0{*}{*}{*}{*})_B = W_2(B^2 + B) \pm W_1B + W_0(B+1)$, where the sign of W_1 is chosen as in (2). A similar method also works nicely when n is odd, if W_2 and W_1 are treated as $(n+1)/2$-digit products and W_0 as an $(n-1)/2$-digit product. (Two digits of extra memory are needed when $n = 3$.)

Page 589 new answer ——— 10 Sep 1995

17. Let $m = s^2 + r$ where $-s < r \le s$. We can use (2) with $U_1 = \lfloor u/s \rfloor$, $U_0 = u \bmod s$, $V_1 = \lfloor v/s \rfloor$, $V_0 = v \bmod s$, and with s playing the role of 2^n. If we know the signs of $U_1 - U_0$ and $V_1 - V_0$ we know how to compute the product $|U_1 - U_0|\,|V_1 - V_0|$, which is $< m$, and whether to add or subtract it. It remains to multiply by s and by $s^2 \equiv -r$. Each of these can be done with four multiplication/divisions, using exercise 3.2.1.1–9, but only seven are needed because one of the multiplications needed to compute $sx \bmod m$ is by r or $r + s$. Thus 14 multiplication/divisions are sufficient (or 12, in case $u = v$ or u is constant).

Page 590 line 10 ——— 20 Apr 1995

[to appear]. $\rightsquigarrow$ [*SIGPLAN Notices* **25**, 6 (June 1990), 112–126]. See also D. E. Knuth in *Beauty is Our Business* (Springer, 1990), 233–242.

Page 592 new answer ——— 20 Apr 1995

17. See W. D. Clinger, *SIGPLAN Notices* **25**, 6 (June 1990), 92–101, and the references in exercise 3.

▶**Page 592** line 1 of answer 4 ——— 01 Jun 1997

$gcd(u, v) \rightsquigarrow \gcd(u, v)$

▶**Page 593** line 1 of answer 13 ——— 26 Jan 1986

$\lfloor n/a \rfloor - \lfloor a/n \rfloor \rightsquigarrow \lfloor b/n \rfloor - \lfloor a/n \rfloor$

▶**Page 594** line 2 of answer 8 ——— 16 Jun 1997

odd $\rightsquigarrow$ even

Page 595 replacement for second paragraph of answer 14 ——— 11 Oct 1995

Further details can be found in exercise 4.5.3–45.

▶**Page 596** line 5 of answer 18 ——— 16 Jun 1997

$1/2^n \rightsquigarrow 1/2^{n-2}$

▶**Page 597** answer 23 ——— 16 Jun 1997

line 1: first sum $\rightsquigarrow$ second sum
line 6: $\gamma - \alpha \rightsquigarrow \gamma - \alpha - \beta n$
line 7: $-\beta n \rightsquigarrow$
line 10: other sum. $\rightsquigarrow$ other sums.

Page 598 replacement for answer 28 ——— 02 Apr 1992

28. (Solution by J. O. Shallit.) This is a problem where it appears to be necessary to prove *more* than was asked just to prove what was asked. Let $S(u, v)$ be the number of subtraction steps taken by Algorithm B on inputs u and v. We will prove that $S(u, v) \le \lg(u + v)$. This will imply that $S(u, v) \le \lfloor \lg(u + v) \rfloor \le \lfloor \lg 2\max(u, v) \rfloor = 1 + \lfloor \lg \max(u, v) \rfloor$ as desired.

Notice that $S(u, v) = S(v, u)$. If u is even, $S(u, v) = S(u/2, v)$; hence we may assume that u and v are odd. We may also assume that $u > v$, since $S(u, u) = 1$. Then $S(u, v) = 1 + S\bigl((u - v)/2, v\bigr) \le 1 + \lg\bigl((u - v)/2 + v\bigr) = \lg(u + v)$ by induction.

It follows, incidentally, that the smallest case requiring n subtraction steps is $u = 2^{n-1} + 1$, $v = 2^{n-1} - 1$.

Page 598 answer 29 — 30 Oct 1995

(New York: Chelsea, 1952) $\rightsquigarrow$ (Carnegie Inst. of Washington, 1919)

Page 599 new paragraph for end of answer 34 — 22 Jul 1996

A possibly faster binary algorithm has been described by J. Sorenson, *J. Algorithms* **16** (1994), 110–144; Shallit and Sorenson, *Lecture Notes in Comp. Sci.* **877** (1994), 169–183.

▸**Page 599** line 2 of step Y6 — 07 Sep 1989

back to B3 $\rightsquigarrow$ back to Y3

Page 599 new paragraph for end of answer 35 — 22 Aug 1996

See also A. W. Bojanczyk and R. P. Brent, *Computers and Math.* **14** (1987), 233, for a similar extension of the algorithm in exercise 37.

Page 599 new answer — 07 Aug 1995

37. Let $n = \lg\max(|u|, |v|)$. We can show inductively that $|u| \le 2^{n-(m-c)/2}$, $|v| \le 2^{n-(m+c)/2}$ after we have performed the operation $c \leftarrow c + 1$ in step C3 m times. Therefore $m \le 2n$. If C2 is executed t times, we have $t \le m + 2$, because m increases every time except the first and last. [See *VLSI '83* (North-Holland, 1983), 145–154.]

Notes: When $u = 1$ and $v = 3 \cdot 2^k - 1$ and $k \ge 2$, we have $n = k + 2$, $m = 2k$, $t = k + 4$. When $u = u_j$ and $v = 2u_{j-1}$ in the sequence defined by $u_0 = 3$, $u_1 = 1$, $u_{j+1} = \min(|3u_j - 16u_{j-1}|, |5u_j - 16u_{j-1}|)$, we have $m = 2j + 2$, $t = 2j + 3$, and (empirically) $n \approx 1.62j$. Can t be asymptotically larger than $2n/1.62$?

▸**Page 600** answer 10 — 23 Oct 1983

line 4: $B_0 = -2$, $B_2 = A_1 - 2 \rightsquigarrow B_0 = -2$, $B_1 = 1$, $B_2 = A_1 - 2$
line 6: $B_7 = A_4$. $\rightsquigarrow B_7 = A_4$, $m = 7$.

Page 600 line 9 of answer 11 — 28 Jan 1997

Note: $\rightsquigarrow$ J.-A. Serret, *J. de Math. Pures et Appl.* **15** (1850), 153–155.

Page 601 new paragraph for end of answer 12 — 04 Jan 1997

This theory of continued fractions for quadratics is largely due to Euler [*Novi Comment. Acad. Sci. Petrop.* **11** (1765), 28–66] and Lagrange [*Hist. Acad. Sci.* **24** (Berlin, 1768), 111–180]. Some of its aspects had been discovered in Japan before 1750; see Y. Mikami, *The Development of Mathematics in China and Japan* (1913), 223–229.

▸**Page 601** line 9 of answer 15 — 15 Apr 1983

$X_k \leftarrow \lfloor A/C \rfloor \rightsquigarrow X_k \leftarrow \min(\lfloor A/C \rfloor, \lfloor (A+B)/(C+D) \rfloor)$

▸**Page 603** middle — 19 Jun 1997

line 16: $x \rightsquigarrow z$ [twice]
line 20: $|z| \le 1 \rightsquigarrow |2z| \le 1$

Page 603 bottom line — 08 Jul 1996

198. $\rightsquigarrow$ 198. The continued fraction for e was first found empirically by Roger Cotes, *Philosophical Transactions* **29** (1714), 5–45, Proposition 1, Scholium 3.

Page 605 answer 30 ______ 11 Oct 1995

lines 11–14: prime to n. Kronecker ... further results ⇝ prime to n.

K. Vahlen [*Crelle* **115** (1895), 221–233] considered all algorithms that replace (u, v) by $(v, (\pm u) \bmod v)$ at each iteration when $u \bmod v \neq 0$. If $u \perp v$ there are exactly v such algorithms, and they can be represented as a binary tree with v leaves. The shallowest leaves, which correspond to the shortest possible number of iterations over all such gcd algorithms, occur when the least remainder is taken at each step; the deepest leaves occur when the greatest remainder is always chosen. [Similar ideas had been considered by Lagrange in *Hist. Acad. Sci.* **23** (Berlin, 1768), 111–180, §58.] For further results

Page 605 answer 31 ______ 11 Oct 1995

lines 3 and 4: $n \geq 2$. ... result ⇝ $n \geq 2$. This result
lines 5–7: Binet. See P. Bachmann, ... algorithm. ⇝ Binet.

Page 606 answer 35 ______ 07 Oct 1995

4722. ⇝ 4722. M. L. V. Pitteway and C. M. A. Castle, *Bull. Inst. Math. and its Applications* **24** (1988), 17–20, have found strong empirical evidence that the sum of all partial quotients is actually

$$\frac{\pi^2}{24(\ln 2)^2}\left(T_n + \frac{1}{2} - \frac{18(\ln 2)^2}{\pi^2}\right)^2 + \frac{6}{\pi^2}\sum_{\substack{p \text{ prime}\\ p^r \backslash n}}\left(\frac{4r}{p^r} - \frac{p+1}{p^{2r}}\frac{p^r-1}{p-1}\right)(\ln p)^2 \\ - 2.542875 + O(n^{-1/2}).$$

▸**Page 606** answer 36 ______ 18 Jun 1997

line 2: $u_k = F_{k_1} \ldots F_{k_{i-1}} F_{k_i} - 1$ ⇝ $u_i = F_{k_1} \ldots F_{k_{i-1}} F_{k_i - 1}$
line 3: positive ⇝ ≥ 2
line 4: $k_{n-1} = 2$, $u_n = F_{N-n+3}$ ⇝ $k_{n-2} = 2$, $u_n = 2F_{N-n+2}$

Page 607 lines 2–8 ______ 11 Apr 1996

Stern, ... D. H. Lehmer, ⇝ Stern, *Crelle* **55** (1858), 193–220, although the relation to binary trees is not explicit in his paper. The notion of obtaining all possible fractions by successively interpolating $(p+p')/(q+q')$ between adjacent elements p/q and p'/q' goes back much further: The essential ideas were published by Daniel Schwenter [*Deliciæ Physico-Mathematicæ* (Nürnberg, 1636), Part 1, Problem 87; *Geometria Practica*, 3rd edition (1641), 68; see M. Cantor, *Geschichte der Math.* **2** (1900), 763–765], and by John Wallis in his *Treatise of Algebra* (1685), Chapters 10–11. C. Huygens put such ideas to good use when designing the gear-wheels of his planetarium [see *Descriptio Automati Planetarii* (1703), published after his death]. Lagrange gave a full description in *Hist. Acad. Sci.* **23** (Berlin, 1767), 311–352, §24, and in his additions to the French translation of Euler's algebra (1774), §18–§20. See also exercise 1.3.2–19; A. Brocot, *Revue Chronométrique* **6** (1860), 186–194; D. H. Lehmer,

▸**Page 607** line −8 ______ 07 Sep 1989

on $m \bmod 4$ ⇝ and on $n \bmod 4$

Page 608 line 6 ______ 06 Jan 1997

father of x in the Stern-Peirce ⇝ parent of x in the Stern-Brocot

Page 608 replacement for answer 45 ———— 11 Oct 1995

45. $T_j = \big(K_{j-2}(-a_2,\ldots,-a_{j-1}),\ K_{j-1}(-a_1,\ldots,-a_{j-1}),\ K_{n-j}(a_{j+1},\ldots,a_n)d\big) = \big((-1)^j K_{j-2}(a_2,\ldots,a_{j-1}), (-1)^{j-1}K_{j-1}(a_1,\ldots,a_{j-1}), K_{n-j}(a_{j+1},\ldots,a_n)d\big)$.

46. Since $\lambda x_1 + \mu z_1 = \mu v$ and $\lambda x_{n+1} + \mu z_{n+1} = -\lambda v/d$, there is an odd value of j such that $\lambda x_j + \mu z_j \ge 0$ and $\lambda x_{j+2} + \mu z_{j+2} \le 0$. If $\lambda x_j + \mu z_j > \theta$ and $\lambda x_{j+2} + \mu z_{j+2} < -\theta$ we have $\mu > \theta/z_j$ and $\lambda > -\theta/x_{j+2}$. It follows that $0 < \lambda x_{j+1} + \mu z_{j+1} < \lambda\mu x_{j+1}z_j/\theta - \lambda\mu z_{j+1}x_{j+2}/\theta \le 2\lambda\mu v/\theta = 2\theta$, because we have $|x_{k+1}z_k| = K_{k-1}(a_2,\ldots,a_k)K_{n-k}(a_{k+1},\ldots,a_n) \le K_{n-1}(a_2,\ldots,a_n) = v/d$ for all k. [H. W. Lenstra, *Math. Comp.* **42** (1984), 331–340.]

Page 608 new answer ———— 29 Aug 1996

47. If $ax - mz = y$ and $x \perp y$ we have $x \perp mz$. Consider the Stern-Brocot tree of exercise 40, with an additional node labeled 0/1. In addition to the node labels z/x, add the tag value $ax - mz$. We want to find all nodes z/x whose tag is $\le \theta = \sqrt{m/2}$ and whose denominator x is also $\le \theta$. The only possible path to such nodes keeps a positive tag to the left and a negative tag to the right. Let z_1/x_1 be the first node whose tag y_1 satisfies $|y_1| \le \theta$. If $x_1 > \theta$, there is no solution, since subsequent values on the path have even larger denominators. Otherwise $(\pm x_1, \mp y_1)$ is a solution, provided that $x_1 \perp y_1$.

If one solution is found, let the previous node z_0/x_0 have tag y_0, where $y_0 > \theta$ and $y_0y_1 < 0$. The next possibilities for a solution are the nodes $(z_0 + kz_1)/(x_0 + kx_1)$ with tags $y_0 + ky_1$, for $k = 1, 2, \ldots$. Notice that the path we have followed corresponds to Algorithm 4.5.2X; hence we have $x_0 = K(a_1,\ldots,a_{j-1})$, $y_0 = \pm K(a_{j+1},\ldots,a_n)d$, $x_1 = K(a_1,\ldots,a_j)$, and $y_1 = \mp K(a_{j+2},\ldots,a_n)d$ by exercise 45, for some j, when $a/m = /\!/a_1,\ldots,a_n/\!/$ and $d = \gcd(a,m)$. Therefore $x_0 + kx_1$ and $|y_0 + ky_1|$ cannot both be $\le \theta$; if they were, we would have $m = K(a_1,\ldots,a_n)d = (x_0+kx_1)|y_1| + x_1|y_0+ky_1| \le \theta^2 + \theta^2 = m$, and equality cannot hold because $m > 2$. When $k = a_{j+1}$ we have $|y_0 + ky_1| = K(a_{j+3},\ldots,a_n)d < y_1 \le \theta$, hence $x_0 + kx_1 > \theta$ and no further solutions are possible.

This discussion proves that the problem can be solved efficiently by applying Algorithm 4.5.2X with $u = m$ and $v = a \bmod m$, but with the following replacement for step X2: "If $v_3 \le \sqrt{m/2}$, the algorithm terminates. The pair $(x,y) = \big(|v_2|, v_3\,\mathrm{sign}(v_2)\big)$ is then the unique solution, provided that $x \perp y$ and $x \le \sqrt{m/2}$; otherwise there is no solution." [P. S. Wang, *Lecture Notes in Comp. Sci.* **162** (1983), 225–235; P. Kornerup and R. T. Gregory, *BIT* **23** (1983), 9–20.]

A similar method works if we require $x < \theta_1$ and $|y| < \theta_2$, whenever $2\theta_1\theta_2 \le m$.

▸**Page 608** line 3 of answer 4 ———— 21 Jun 1997

$2^{\lceil \lg \max(l-\lambda,\lambda)\rceil} \rightsquigarrow 2^{\lceil \lg \max(l+1-\lambda,\lambda)\rceil}$

Page 608 answer 4 ———— 11 Nov 1980

lines -2 and -1: $p = 2.4771 \rightsquigarrow p \approx 2.4771366$
line -1: .] $\rightsquigarrow$. See *BIT* **20** (1980), 176–184.]

▸**Page 609** line 3 of answer 8 ———— 26 Jan 1986

$p \leftarrow 1,\ p \leftarrow 3 \rightsquigarrow p \leftarrow 3$

Page 609 last paragraph of answer 8 ———— 20 Dec 1996

Improvements $\rightsquigarrow$ The original sieve of Eratosthenes was described in Book 1, Chapter 13 of Nicomachus's *Introduction to Arithmetic*. Improvements

Page 609 at the end of answer 9 ———— 30 Oct 1995

107.] $\leadsto$ 107. Alford, Granville, and Pomerance proved in *Annals of Math.* (2) **139** (1994), 703–722, that there are at least $\Omega(N^{.29})$ Carmichael numbers less than N.]

▶**Page 610** answer 12 ———— 10 Jun 1987

line 2: single-precision integers $\leadsto$ integers
step F2: Set $k \leftarrow 0$. $\leadsto$ Set $k \leftarrow m$.
step F3: If $k > m$ $\leadsto$ If $k < 0$
steps F3 and F4: set $k \leftarrow k + 1$ $\leadsto$ set $k \leftarrow k - 1$
step F4: e_k.) $\leadsto$ e_k. The values E_{jr} are not guaranteed to remain single-precision, but they tend to remain small when k decreases from m to 0 as recommended by Morrison and Brillhart.)

▶**Page 611** line 4 ———— 21 Jun 1997

$V_p \equiv P^n \leadsto V_p \equiv P^p$

Page 611 line −6 ———— 06 May 1995

Golomb's constant λ $\leadsto$ the Dickman-Golomb constant .62433

Page 612 new paragraph to follow answer 19 ———— 21 Oct 1995

An analogous method that finds prime factors p for which $p + 1$ has only small prime divisors is discussed by H. C. Williams, *Math. Comp.* **39** (1982), 225–234.

Page 612 new answer ———— 14 Apr 1996

21. Some interesting theory relevant to this conjecture has been introduced by Eric Bach, *Information and Computation* **90** (1991), 139–155.

Page 612 line −7 ———— 20 Mar 1995

(1980), to appear $\leadsto$ **12** (1980), 128–138

Page 613 just before answer 24 ———— 08 May 1995

[Univ. de Paris-Sud ... (1978).] $\leadsto$ [*Theoretical Comp. Sci.* **12** (1980), 97–108.]

▶**Page 613** answer 24 ———— 17 Jan 1981

line 3: passes $\leadsto$ fails
lines 4 and 5: at most $\frac{1}{4}qn + \cdots < \frac{1}{2}N$ $\leadsto$ at most $-1 + q(b_n + 1) + \min(b_n + 1, r) \le q\bigl(\frac{1}{4}(n-1) + 1\bigr) + \min(b_n + 1, r) < \frac{1}{3}qn + \min(\frac{1}{4}n, r) = \frac{1}{3}N + \min(\frac{1}{4}n - \frac{1}{3}r, \frac{2}{3}r) \le \frac{1}{3}N + \frac{1}{6}n \le \frac{1}{2}N$

▶**Page 614** top ———— 24 Oct 1995

line 1: to ... The constant $\leadsto$ to $C + \int_x^\infty dt/\bigl((t^2 - 1)t\ln t\bigr)$, minus $\frac{1}{2k}$ if n is the kth power of a prime. The constant C

Page 614 additional material for answer 25 ———— 08 Oct 1995

[For a summary of developments during the 100 years following von Mangoldt's paper, see A. A. Karatsuba, *Complex Analysis in Number Theory* (CRC Press, 1995). See also Eric Bach and Jeffrey Shallit, *Algorithmic Number Theory* **1** (MIT Press, 1996), Chapter 8, for an excellent introduction to the connection between Riemann's hypothesis and concrete problems about integers.]

Page 614 answer 27 — 25 Feb 1995

line 4: $k \cdot 2^n + 1 \bmod 12 \rightsquigarrow (k \cdot 2^n + 1) \bmod 12$
line −4: 3912 $\rightsquigarrow$ 3912, 20909, 34350, 42294, 42665, 44685, 48150, 55182, 59973, 80190, 157169, 213321, and no other $n \le 300000$
last three lines: 127, 1947, ... Plass.] $\rightsquigarrow$ 127, 1947, 3313, 4687, 5947, 13165, 23473, 26607, 125413, 209787, 240937, and no other $n \le 300000$. See R. M. Robinson, *Proc. Amer. Math. Soc.* **9** (1958), 673–681; G. V. Cormack and H. C. Williams, *Math. Comp.* **35** (1980), 1419–1421; H. Dubner and W. Keller, *Math. Comp.* **64** (1995), 397–405; J. S. Young, *Math. Comp.*, to appear.]

▸**Page 614** bottom line — 21 Jun 1997

$(r/2)! \rightsquigarrow (r/2)!^2$

Page 614 bottom line — 04 Jul 1982

J. Number Theory, to appear. $\rightsquigarrow$ *J. Algorithms* **3** (1982), 101–127, where Schnorr presents many further refinements of Theorem D.

Page 615 answer 31 — 16 May 1996

$r = M \rightsquigarrow n = M$

▸**Page 615** answer 34 — 06 May 1995

line 3: $\ge \frac{1}{2}\max(P,Q) \ge \frac{1}{2}10^{-6} \rightsquigarrow \ge 2(10^{-6} - 10^{-12})$
line 8: with high probability we find $\rightsquigarrow$ we find as in Algorithm P
line 9: as in Algorithm P $\rightsquigarrow$ with probability $\ge \frac{1}{2}$

▸**Page 615** line 2 of answer 35 — 13 Oct 1984

$\left(\frac{2}{p}\right) = -\left(\frac{2}{q}\right) = 1 \rightsquigarrow \left(\frac{2}{p}\right) = -\left(\frac{2}{q}\right) = -1$

▸**Page 615** lines 11 and 12 of answer 35 — 26 Jan 1986

a number that has $\rightsquigarrow$ and $\bigl((\pm f) \bmod N\bigr) - 1$ has

Page 615 additional line at end of answer 35 — 20 Mar 1995

[*Reference: IEEE Transactions on Information Theory* **IT-26** (1980), 726–729.]

▸**Page 615** bottom line — 30 Sep 1982

the number $\rightsquigarrow$ the square root of the number

▸**Page 616** answer 38 — 26 Jan 1986

line 5: $A \leftarrow 2^{2^{l+1}} \rightsquigarrow A = 2^{2^{l+1}}$
line 24: $\binom{2(m+s)}{m+s} = 0 \rightsquigarrow \binom{2(m+s)}{m+s} \bmod n = 0$

Page 616 new answer — 05 Apr 1981

39. After finding $a^{n_i} \bmod p = \prod_{j=1}^m p_j^{e_{ij}}$ for enough n_i, we can solve $\sum_i x_{ijk}e_{ij} + (p-1)t_{jk} = \delta_{jk}$ in integers x_{ijk}, t_{jk} for $1 \le j,k \le m$ (e.g., as in 4.5.2–(23)), thereby knowing the solutions $N_j = \bigl(\sum_i x_{ijk}e_{jk}\bigr) \bmod (p-1)$ to $a^{N_j} \bmod p = p_j$. Then if $ba^{n'} \bmod p = \prod_{j=1}^m p_j^{e'_j}$, we have $n + n' \equiv \sum_{j=1}^m e'_j N_j$ (modulo p). [Cf. *FOCS* **20** (1979), 55–60. Improved algorithms have been found by Coppersmith, Odlyzko, and Schroeppel, *Algorithmica* **1** (1986), 1–15.]

Page 616 new answer ______ 19 Apr 1995

40. Let $r = (10^{50}-1)/9$. (a) $10^{49}+9$. (b) $r+3\cdot 10^{46}$. (c) $2r+3\cdot 10^{47}+7$. (d) $3r+2\cdot 10^{49}$. (e) $4r+2\cdot 10^{49}-3$. (f) $5r+3\cdot 10^{49}+4$. (g) $6r+2\cdot 10^{48}+3$. (h) $7r+2\cdot 10^{25}$. (i) $8r+10^{38}-7$. (j) $9r-8000$.

Page 616 new answer ______ 10 Sep 1995

41. Notice that it's easy to prove the primality of q when $q-1$ has just 2 and p as prime factors. The only successors of 2 are Fermat primes, and the existence or nonexistence of a sixth Fermat prime is one of the most famous unsolved problems of number theory. Thus we probably will never know how to determine whether or not an arbitrary integer has any successors. In some cases, however, this is possible; for example, John Selfridge proved in 1962 that 78557 and 271129 have none [see *AMM* **70** (1963), 101–102], after W. Sierpiński had proved the existence of infinitely many odd numbers without a successor [*Elemente der Math.* **15** (1960), 73–74]. Perhaps 78557 is the smallest of these, although 69 other contenders for that honor still existed in 1983, according to G. Jaeschke and W. Keller [*Math. Comp.* **40** (1983), 381–384, 661–673; **45** (1985), 637].

For information on the more traditional "Cunningham" form of prime chain, in which the transitions are $p \to 2p\pm 1$, see Günter Löh, *Math. Comp.* **53** (1989), 751–759. In particular, Löh found that $554688278430 \cdot 2^k - 1$ is prime for $0 \le k < 12$.

Page 616 new answer ______ 11 Oct 1995

42. L1. [Initialize.] Find $\bar r$ such that $r\bar r \equiv 1$ (modulo s); then set $r' \leftarrow n\bar r \bmod s$, $u \leftarrow r'\bar r \bmod s$, $v \leftarrow s$, $w \leftarrow (n - rr')\bar r/s \bmod s$, $\theta \leftarrow \lfloor\sqrt{N/s}\rfloor$, $(u_1, u_3) \leftarrow (1, u)$, $(v_1, v_3) \leftarrow (0, v)$. (We want to find all pairs of integers (λ, μ) such that $(\lambda s + r)(\mu s + r') = N$; this implies $\lambda u + \mu \equiv w$ (modulo s) and $\sqrt{\lambda\mu v} \le \theta$. We will perform Algorithm 4.5.2X with t_2, u_2, v_2 suppressed; the relations

$$\lambda t_3 + \mu t_1 \equiv w t_1, \quad \lambda u_3 + \mu u_1 \equiv w u_1, \quad \lambda v_3 + \mu v_1 \equiv w v_1 \qquad (\text{modulo } s)$$

will remain invariant.)

L2. [Try for divisors.] If $v_1 = 0$, output $\lambda s + r$ whenever $\lambda s + r$ divides N and $0 \le \lambda \le \theta/s$. If $v_3 = 0$, output $N/(\mu s + r')$ whenever $\mu s + r'$ divides N and $0 \le \mu \le \theta/s$. Otherwise, for all k such that $|wv_1 + ks| \le \theta$ if $v_1 < 0$, or $0 < wv_1 + ks \le 2\theta$ if $v_1 > 0$, and for $\sigma = +1$ and -1, output $\lambda s + r$ if $d = (wv_1 s + ks^2 + v_3 r + v_1 r')^2 - 4v_1 v_3 N$ is a perfect square and if the numbers

$$\lambda = \frac{wv_1 s + ks^2 - v_3 r + v_1 r' + \sigma\sqrt{d}}{2v_3 s}, \quad \mu = \frac{wv_1 s + ks^2 + v_3 r - v_1 r' - \sigma\sqrt{d}}{2v_3 s}$$

are positive integers. (These are the solutions to $\lambda v_3 + \mu v_1 = wv_1 + ks$, $(\lambda s + r)(\mu s + r') = N$.)

L3. [Done?] If $v_3 = 0$, the algorithm terminates.

L4. [Divide and subtract.] Set $q \leftarrow \lfloor u_3/v_3 \rfloor$. If $u_3 = qv_3$ and $v_1 < 0$, decrease q by 1. Then set

$$(t_1, t_3) \leftarrow (u_1, u_3) - (v_1, v_3)q, \quad (u_1, u_3) \leftarrow (v_1, v_3), \quad (v_1, v_3) \leftarrow (t_1, t_3)$$

and return to step L2. ▌

[Cf. *Math. Comp.* **42** (1984), 331–340. The bounds in step L2 can be sharpened, for example to ensure that $d \ge 0$. Some factors may be output more than once.]

Page 616 new answer ________ 09 Jan 1997

43. (a) First make sure that the Jacobi symbol $\left(\frac{y}{m}\right)$ is $+1$. (If it's 0, the task is easy; if it's -1, then $y \notin Q_m$.) Then choose random integers $x_1, \ldots, x_n$ in $[0\,.\,.\,m)$ and let $X_j = \bigl[G(y^2x_j^4 \bmod m) = (yx_j^2 \bmod m) \bmod 2\bigr]$. If $y \in Q_m$ we have $\mathrm{E}\,X_j \geq \frac{1}{2} + \epsilon$; otherwise $m - y \in Q_m$ and $\mathrm{E}\,X_j \leq \frac{1}{2} - \epsilon$. Report that $y \in Q_m$ if $X_1 \cdots + X_n \geq \frac{1}{2}n$. The probability of failure is at most $e^{-2\epsilon^2 n}$, by exercise 1.2.10–22. Therefore we choose $n = \lceil \frac{1}{2}\epsilon^{-2} \ln \delta^{-1} \rceil$.

(b) Find an x with Jacobi symbol $\left(\frac{x}{m}\right) = -1$, and set $y \leftarrow x^2 \bmod m$. Then the prime factors of m are $\gcd(x + \sqrt{y}, m)$ and $\gcd(x - \sqrt{y}, m)$, so our task is to find $\pm\sqrt{y}$ when $y \in Q_m$ is given. If we can find τv for any nonzero v, we are done, since $\sqrt{y} = (v^{-1}\tau v) \bmod m$ unless $\gcd(v, m)$ is a factor of m.

Assume that $\epsilon = 2^{-e}$ for some $e \geq 1$. Choose random integers a and b in $[0\,.\,.\,m)$, and assume that we know the binary fractions α_0 and β_0 such that

$$\left|\frac{\tau a}{m} - \alpha_0\right| < \frac{\epsilon}{64}, \qquad \left|\frac{\tau b}{m} - \beta_0\right| < \frac{\epsilon^3}{64};$$

here α_0 is an odd multiple of $\epsilon/64$, while β_0 is an odd multiple of $\epsilon^3/64$. Assume also that we know λa and λb. Of course we don't really know α_0, β_0, λa, or λb, but we will try all $32\epsilon^{-1} \times 32\epsilon^{-3} \times 2 \times 2$ possibilities. Spurious branches of the program, which operate under incorrect assumptions, will cause no harm.

Define the numbers $u_{tj} = 2^{-t}\bigl(a + (j + \frac{1}{2})b\bigr) \bmod m$ and $v_{tj} = 2^{-t-1}(a + jb) \bmod m$. Both u_{tj} and v_{tj} are uniformly distributed in $[0\,.\,.\,m)$, because a and b were chosen at random. Furthermore, for fixed t, the numbers u_{tj} for $j_0 \leq j < j_0 + l$ are *pairwise independent*, and so are the numbers v_{tj} for $j_0 \leq j < j_0 + l$, as long as l does not exceed the smallest prime factor of m. We will make use of u_{tj} and v_{tj} only for $-2r\epsilon^{-2} \leq j < 2r\epsilon^2$; if any of these values has a nonzero factor in common with m, we're done.

For all $v \perp m$ we define $\chi v = +1$ if $v \in Q_m$, $\chi v = -1$ if $-v \in Q_m$, and $\chi v = 0$ if $\left(\frac{v}{m}\right) = -1$. Notice that $\chi u_{(t+2)j} = \chi u_{tj}$, since $u_{tj} = (2^2 u_{(t+2)j}) \bmod m$. Therefore we can determine χu_{tj} and χv_{tj} for all t and j by applying algorithm A to u_{tj} and v_{tj} for $0 \leq t \leq 1$ and $-2r\epsilon^{-2} \leq j < 2r\epsilon^{-2}$. Setting $\delta = \frac{1}{1440}\epsilon^2 r^{-1}$ in that algorithm will ensure that all χ values are correct with probability $\geq 1 - \frac{1}{90}$.

The algorithm works in at most r stages. At the beginning of stage t, for $0 \leq t < r$, we assume that we know $\lambda 2^{-t}a$, $\lambda 2^{-t}b$, and fractions α_t, β_t such that

$$\left|\frac{\tau 2^{-t}a}{m} - \alpha_t\right| < \frac{\epsilon}{2^{t+6}}, \qquad \left|\frac{\tau 2^{-t}b}{m} - \beta_t\right| < \frac{\epsilon^3}{2^{t+6}}.$$

Define $\alpha_{t+1} = \frac{1}{2}(\alpha_t + \lambda 2^{-t}a)$ and $\beta_{t+1} = \frac{1}{2}(\beta_t + \lambda 2^{-t}b)$; this preserves the inequalities. The next step is to find $\lambda 2^{-t-1}b$, which satisfies

$$\lambda u_{tj} + \lambda 2^{-t}a + j\lambda 2^{-t}b + \lambda 2^{-t-1}b + \left\lfloor \frac{\tau 2^{-t}a + j\tau 2^{-t}b + \tau 2^{-t-1}b}{m} \right\rfloor \equiv 0 \quad (\text{modulo } 2)\,.$$

Let $n = 4\min(r, 2^t)\epsilon^{-2}$; then when $|j| \leq \frac{n}{2}$ we have

$$\left|\frac{\tau 2^{-t}a}{m} + j\frac{\tau 2^{-t}b}{m} + \frac{\tau 2^{-t-1}b}{m} - (\alpha_t + j\beta_t + \beta_{t+1})\right| < \frac{\epsilon}{16}.$$

Therefore if $\chi u_{tj} = 1$ it is likely that $\lambda 2^{-t-1}b = G_j$, where $G_j = \bigl(G(u_{tj}^2 y \bmod m) + \lambda 2^{-t}a + j\lambda 2^{-t}b + \lfloor \alpha_t j\beta_t + \beta_{t+1} \rfloor\bigr) \bmod 2$. More precisely, we will have $\lfloor(\tau 2^{-t}a + j\tau 2^{-t}b +$

$\tau 2^{-t-1}b)/m\rfloor = \lfloor \alpha_t + j\beta_t + \beta_{t+1}\rfloor$ unless $\tau u_{tj} < \frac{\epsilon}{16}m$ or $\tau u_{tj} > \bigl(1-\frac{\epsilon}{16}\bigr)m$. Let $Y_j = (2G_j - 1)\chi u_{tj}$. If $Y_j = +1$, it is a vote for $\lambda 2^{-t-1}b = 1$; if $Y_j = -1$, it is a vote for $\lambda 2^{-t-1}b = 0$; if $Y_j = 0$, it is an abstention. We will be democratic and set $\lambda 2^{-t-1}b = \bigl[\sum_{j=-n/2}^{n/2-1} Y_j \ge 0\bigr]$.

What is the probability that $\lambda 2^{-t-1}b$ is correct? Let $Z_j = -1$ if $\chi u_{tj} \ne 0$ and $(\tau u_{tj} < \frac{\epsilon}{16}m$ or $\tau u_{tj} > \bigl(1-\frac{\epsilon}{16}\bigr)m$ or $G(u_{tj}^2 y \bmod m) \ne \lambda u_{tj})$; otherwise let $Z_j = |\chi u_{tj}|$. Since Z_j is a function of u_{tj}, the random variables Z_j are pairwise independent and have the same distribution. Let $Z = \sum_{j=-n/2}^{n/2-1} Z_j$; if $Z > 0$, the value of $\lambda 2^{-t-1}b$ will be correct. The probability that $Z_j = 0$ is $\frac{1}{2}$, and the probability that $Z_j = +1$ is $\ge \frac{1}{4} + \frac{\epsilon}{2} - \frac{\epsilon}{8}$; therefore $\mathrm{E}\,Z_j \ge \frac{3}{4}\epsilon$. Clearly $\mathrm{var}(Z_j) \le \frac{1}{2}$. So the chance of error, in the branch of the program that has the correct assumptions, is at most $\Pr(Z \le 0) \le \Pr\bigl((Z - n\,\mathrm{E}\,Z_j)^2 \ge \frac{9}{16}n^2\epsilon^2\bigr) \le \frac{8}{9}n^{-1}\epsilon^2 = \frac{2}{9}\min(r, 2^t)^{-1}$, by Chebyshev's inequality (exercise 3.5–42).

A similar method, with v_{tj} in place of u_{tj}, can be used to determine $\lambda 2^{-t-1}a$ with error $\le \frac{2}{9}\min(r, 2^t)^{-1}$. Eventually we will have $\epsilon^3/2^{t+6} < 1/(2m)$, so $\tau 2^{-t}b$ will be the nearest integer to $m\beta_t$. Then we can compute $\sqrt{y} = (2^t b^{-1}\tau 2^{-t}b) \bmod m$; squaring this quantity will tell us if we are correct.

The total chance of making a mistake is bounded by $\frac{4}{9}\sum_{t\ge 1} 2^{-t} = \frac{4}{9}$ in stages $t < \lg n$, and by $\frac{4}{9}\sum_{t\le r} r^{-1} = \frac{4}{9}$ in subsequent stages. So the total chance of error, including the possibility that the χ values were not all correct, is at most $\frac{4}{9}+\frac{4}{9}+\frac{1}{90} = \frac{9}{10}$. At least $\frac{1}{10}$ of all runs of the program will succeed in finding $\sqrt{y}$; hence the factors of m will be found after repeating the process at most 10 times, on the average.

The total running time is dominated by $O\bigl(r\epsilon^{-4}\log(r\epsilon^{-2})T(G)\bigr)$ for the χ computation, plus $O\bigl(r^2\epsilon^{-2}T(G)\bigr)$ for subsequent guessing, plus $O\bigl(r^2\epsilon^{-6}\bigr)$ for the calculations of α_t, β_t, $\lambda 2^{-t}a$, and $\lambda 2^{-t}b$ in all branches.

This procedure, which nicely illustrates many of the basic paradigms of randomized algorithms, is due to R. Fischlin and C. P. Schnorr [*Proc. Eurocrypt 97*, to appear], who derived it from earlier approaches by Alexi, Chor, Goldreich, and Schnorr [*SICOMP* **17** (1988), 194–209] and by Ben-Or, Chor, and Shamir [*STOC* **15** (1983), 421–430]. When we combine it with Lemma 3.5P4, we get a theorem analogous to Theorem 3.5P, but with the sequence 3.2.2–(16) instead of 3.2.2–(17). Fischlin and Schnorr showed how to streamline the calculations so that their factoring algorithm takes $O\bigl(r\epsilon^{-4}\log(r\epsilon^{-1})T(G)\bigr)$ steps; the resulting time bound for "cracking" 3.2.2–(16) is $T(F) = O\bigl(RN^4\epsilon^{-4}\log(RN\epsilon^{-1})(T(G) + R^2)\bigr)$. The constant factor implied by this O is rather large, but not enormous. A similar method finds x from the RSA function $y = x^a \bmod m$ when $a \perp \phi(m)$, if we can guess $y^{1/a} \bmod 2$ with probability $\ge \frac{1}{2} + \epsilon$.

Page 616 new answer ———————————————————— 20 Oct 1995

44. Suppose $\sum_{j=0}^{d-1} a_{ij}x^j \equiv 0$ (modulo m_i), $\gcd(a_{i0}, a_{i1}, \ldots, a_{i(d-1)}, m_i) = 1$, and $|x| < m_i$ for $1 \le i \le k = \binom{d}{2} + 1$, where $m_i \perp m_j$ for $1 \le i < j \le k$. Also assume that $m = \min\{m_1, \ldots, m_k\} > n^{n/2}2^{n^2/2}d^d$, where $n = d + k$. First find $u_1, \ldots, u_k$ such that $u_j \bmod m_i = \delta_{ij}$. Then set up the $n \times n$ matrix

$$L = \begin{pmatrix} M \\ 0 & mM \\ \vdots & \vdots & \ddots \\ 0 & 0 & \ldots & m^{d-1}M \\ a_{10}u_1 & ma_{11}u_1 & \ldots & m^{d-1}a_{1(d-1)}u_1 & M/m_1d \\ a_{20}u_2 & ma_{21}u_2 & \ldots & m^{d-1}a_{2(d-1)}u_2 & 0 & M/m_2d \\ \vdots & \vdots & & \vdots & \vdots & \vdots & \ddots \\ a_{k0}u_k & ma_{k1}u_k & \ldots & m^{d-1}a_{k(d-1)}u_k & 0 & 0 & \ldots & M/m_kd \end{pmatrix}$$

where $M = m_1m_2\ldots m_k$; all entries above the diagonal are zero, hence $\det L = M^{n-1}m^{k-1}d^{-k}$. Now let $v = (t_0, \ldots, t_{d-1}, v_1, \ldots, v_k)$ be a nonzero integer vector with $\operatorname{length}(vL) \le \sqrt{n2^n}M^{(n-1)/n}m^{(k-1)/n}d^{-k/n}$. Since $M^{(n-1)/n} < M/m^{k/n}$, we have $\operatorname{length}(vL) < M/d$. Let $c_j = t_jM + \sum_{i=1}^{k} a_{ij}u_iv_i$ and $P(x) = c_0 + c_1x + \cdots + c_{d-1}x^{d-1}$. Then $P(x) \equiv v_i(a_{i0} + a_{i1}x + \cdots + a_{i(d-1)}x^{d-1}) \equiv 0$ (modulo m_i), for $1 \le i \le k$; hence $P(x) \equiv 0$ (modulo M). Also $|m^jc_j| < M/d$; it follows that $P(x) = 0$. But $P(x)$ is not identically zero, because the conditions $v_ia_{ij} \equiv 0$ (modulo m_i) and $\gcd(a_{i0}, \ldots, a_{i(d-1)}, m_i) = 1$ imply $v_i \equiv 0$ (modulo m_i), while $|\,v_iM/m_id\,| < M/d$ implies $|v_i| < m_i$; we cannot have $v_1 = \cdots = v_k = 0$. Thus we can find x (more precisely, at most $d-1$ possibilities for x), and the total running time is polynomial in $\lg M$. [*Lecture Notes in Comp. Sci.* **218** (1985), 403–408.]

Page 616 new answer ______ 21 Aug 1996

45. *Fact 1:* A solution always exists. Suppose first that n is prime. If $\left(\frac{b}{n}\right) = 1$, there is a solution with $y = 0$. If $\left(\frac{b}{n}\right) = -1$, let $j > 0$ be minimum such that we have $\left(\frac{-ja}{n}\right) = -1$; then $x_0^2 - a \equiv -ja$ and $b \equiv -ja(y_0)^2$ for some x_0 and y_0 (modulo n), hence $(x_0y_0)^2 - ay_0^2 \equiv b$. Suppose next that we have found a solution $x^2 - ay^2 \equiv b$ (modulo n) and we want to extend this to a solution modulo n^2. We can always find c and d such that $(x+cn)^2 - a(y+dn)^2 \equiv b$ (modulo n^2), because $(x+cn)^2 - a(y+dn)^2 \equiv x^2 - ay^2 + (2cx - 2ayd)n$ and $\gcd(2x, 2ay) \perp n$. Thus a solution always exists when n is a power of an odd prime. (We need to assume that n is odd because, for example, there is no solution to $x^2 \pm y^2 \equiv 3$ (modulo 8).) Finally, a solution exists for all odd n, by the Chinese remainder theorem.

Fact 2: The number of solutions, given a and n with $a \perp n$, is the same for all $b \perp n$. This follows from the hinted identity and Fact 1, for if $x_1^2 - ay_1^2 \equiv b$ then $(x_1x_2 - ay_1y_2, x_1y_2 + x_2y_1)$ runs through all solutions of $x^2 - ay^2 \equiv b$ as (x_2, y_2) runs through all solutions of $x^2 - ay_2 \equiv 1$. In other words, (x_2, y_2) is uniquely determined by (x_1, y_1) and (x, y), when $x_1^2 - ay_1^2 \perp n$.

Fact 3: Given integers (a, s, z) such that $z^2 \equiv a$ (modulo s), we can find integers (x, y, m, t) with $x^2 - ay^2 = m^2st$, where $(x, y) \ne (0, 0)$ and $t^2 \le \frac{4}{3}|a|$. For if $z^2 = a + ms$, let (u, v) be a nonzero pair of integers that minimizes $(zu + mv)^2 + |a|u^2$. We can find (u, v) efficiently using the methods of Section 3.3.4, and $(zu + mv)^2 + |a|u^2 \le \left(\frac{4}{3}|a|\right)^{1/2}$ by exercise 3.3.4–9. Therefore $(zu + mv)^2 - au^2 = mt$ where $t^2 \le \frac{4}{3}|a|$. The hinted identity now solves $x^2 - ay^2 = (ms)(mt)$.

Fact 4: It is easy to solve $x^2 - y^2 \equiv b$ (modulo n), because we can let $x = (b+1)/2$, $y = (b-1)/2$.

Fact 5: It is not difficult to solve $x^2 + y^2 \equiv b$ (modulo n), because the method in exercise 3.3.4–11 solves $x^2 + y^2 = p$ when p is prime and $p \bmod 4 = 1$; one of the numbers b, $b+n$, $b+2n$, $\ldots$ will be such a prime.

Now to solve the stated problem when $|a| > 1$ we can proceed as follows. Choose u and v at random between 1 and $n-1$, then compute $w = (u^2 - av^2) \bmod n$ and $d = \gcd(w,n)$. If $1 < d < n$ or if $\gcd(v,n) > 1$ we can reduce n; the methods used to prove Fact 1 will lift solutions for factors of n to solutions for n itself. If $d = n$ and $v \perp n$, we have $(u/v)^2 \equiv a$ (modulo n), hence we can reduce a to 1. Otherwise $d = 1$; let $s = bw \bmod n$. This number s is uniformly distributed among the numbers prime to n, by Fact 2. If $\left(\frac{a}{s}\right) = 1$, try to solve $z^2 \equiv a$ (modulo s), assuming that s is prime (exercise 4.6.2–15). If unsuccessful, start over with another random choice of u and v. If successful, let $z^2 = a + ms$ and compute $d = \gcd(ms,n)$. If $d > 1$, reduce the problem as before. Otherwise use Fact 3 to find $x^2 - ay^2 = m^2 st$ with $t^2 \le \frac{4}{3}|a|$; this makes $(x/m)^2 - a(y/m)^2 \equiv st$ (modulo n). If $t = 0$, reduce a to 1. Otherwise apply the algorithm recursively to solve $X^2 - tY^2 \equiv a$ (modulo n). (Since t is much smaller than a, only $O(\log\log n)$ levels of recursion will be necessary.) If $\gcd(Y,n) > 1$ we can reduce n or a; otherwise $(X/Y)^2 - a(1/Y)^2 \equiv t$ (modulo n). Finally the hinted identity yields a solution to ${x'}^2 - a{y'}^2 \equiv s$ (see Fact 2), which leads in turn to the desired solution because $u^2 - av^2 \equiv s/b$.

In practice only $O(\log n)$ random trials are needed before the assumptions about prime numbers made in this algorithm turn out to be true. But a formal proof would require us to assume the Generalized Riemann Hypothesis [*IEEE Trans.* **IT33** (1987), 702–709]. Adleman, Estes, and McCurley [*Math. Comp.* **48** (1987), 17–28] have developed a slower and more complicated algorithm that does not rely on any unproved hypotheses.

▸**Page 618** line 11 ———— 23 Jun 1997

$S_t = \binom{n}{t}(t+1) \leadsto S_t = \binom{n}{t}t$

Page 618 new paragraph for end of answer 6 ———— 02 Sep 1996

There is a similar algorithm analogous to the method of exercise 4.5.2–37. The average number of iterations for both algorithms has been found by G. H. Norton, *SICOMP* **18** (1989), 608–624; K. Ma and J. von zur Gathen, *J. Symbolic Comp.* **9** (1990), 429–455.

Page 619 lines 13–14 of answer 12 ———— 06 Jan 1983

[*SIAM J. Computing*, to appear] $\leadsto$ [*Computing*, Supplement 4 (1982), 173–187]

Page 619 last line of answer 12 ———— 10 Jan 1981

[*JACM*, to appear.] $\leadsto$ [Cf. *JACM* **27** (1980), 701–717.]

▸**Page 620** answer 15 ———— 08 Jun 1983

line 2: replace row i … where $c = c_{ij}/c_{jj}$ $\leadsto$ replace row i and column i by $(c_{i1} - tc_{j1}, \ldots, c_{in} - tc_{jn})$, where $t = c_{ij}/c_{jj}$

line 4: $c_{ii} - c_{ij}^2/c_{ij} \leadsto c_{ii} - c_{ij}^2/c_{jj}$

Page 620 answer 15 ———— 30 Oct 1995

line 3: the determinant $\leadsto$ $\det C$

line 6: "Schmidt's … see *Math.* $\leadsto$ the "Gram-Schmidt orthogonalization process"; see *Crelle* **94** (1883), 41–73; *Math.*

Page 620 replacement for answer 16 ———————— 20 Apr 1995

16. A univariate polynomial of degree d over any unique factorization domain has at most d roots (see exercise 3.2.1.2–16(b)); so if $n=1$ it is clear that $|r(S_1)| \le d_1$. If $n>1$ we have $f(x_1,\ldots,x_n) = g_0(x_2,\ldots,x_n) + x_1 g_1(x_2,\ldots,x_n) + \cdots + x_1^{d_1} g_{d_1}(x_2,\ldots,x_n)$ where g_k is nonzero for at least one k. Given $(x_2,\ldots,x_n)$, it follows that $f(x_1,\ldots,x_n)$ is zero for at most d_1 choices of x_1, unless $g_k(x_2,\ldots,x_n)=0$; hence $|r(S_1,\ldots,S_n)| \le d_1(|S_2|-d_2)\ldots(|S_n|-d_n) + |S_1|\bigl(|S_2|\ldots|S_n| - (|S_2|-d_2)\ldots(|S_n|-d_n)\bigr)$. [R. A. DeMillo and R. J. Lipton, *Inf. Proc. Letters* **7** (1978), 193–195.]

Notes: This upper bound is best possible, because equality occurs for the polynomial $f(x_1,\ldots,x_n) = \prod\{x_j - s_k \mid s_k \in S_j,\ 1 \le k \le d_j,\ 1 \le j \le n\}$. But there's another sense in which the upper bound can be significantly improved: Let $f_1(x_1,\ldots,x_n) = f(x_1,\ldots,x_n)$, and let $f_{j+1}(x_{j+1},\ldots,x_n)$ be any nonzero coefficient of a power of x_j in $f_j(x_j,\ldots,x_n)$. Then we can let d_j be the degree of x_j in f_j instead of the (often much larger) degree of x_j in f. For example, we could let $d_1=3$ and $d_2=1$ in the polynomial $x_1^3x_2^9 - 3x_1^2x_2 + x_2^{100} + 5$. This observation ensures that $d_1+\cdots+d_n \le d$ when each term of f has total degree $\le d$; hence the probability in such cases is

$$\frac{|r(S,\ldots,S)|}{|S|} \le 1 - \left(1-\frac{d_1}{|S|}\right)\ldots\left(1-\frac{d_n}{|S|}\right) \le \frac{d_1+\cdots+d_n}{|S|} \le \frac{d}{|S|}$$

when all sets S_j are equal. If this probability is $\le \frac12$, and if $f(x_1,\ldots,x_n)$ turns out to be zero for 50 randomly selected vectors $(x_1,\ldots,x_n)$, then $f(x_1,\ldots,x_n)$ is identically zero with probability at least $1-2^{-50}$.

Moreover, if $f_j(x_j,\ldots,x_n)$ has the special form $x_j^{e_j} f_{j+1}(x_{j+1},\ldots,x_n)$ with $e_j>0$ we can take $d_j=1$, because x_j must then be 0 when $f_{j+1}(x_{j+1},\ldots,x_n)\ne 0$. A sparse polynomial with only m nonzero terms will therefore have $d_j \le 1$ for at least $n - \lg m$ values of j.

Applications of this inequality to gcd calculation and other operations on sparse multivariate polynomials were introduced by R. Zippel, *Lecture Notes in Comp. Sci.* **72** (1979), 216–226. J. T. Schwartz [*JACM* **27** (1980), 701–717] gave further extensions, including a way to avoid large numbers by means of modular arithmetic: If the coefficients of f are integers, if P is a set of prime numbers all $\ge q$, and if $|f(x_1,\ldots,x_n)| \le L$ whenever each $x_j \in S_j$, then the number of solutions to $f(x_1,\ldots,x_n) \equiv 0$ (modulo p) for $p \in P$ is at most

$$|S_1|\ldots|S_n||P| - (|S_1|-d_1)\ldots(|S_n|-d_n)(|P| - \log_q L).$$

Page 623 new answer ———————— 25 Aug 1996

27. The ideas of answer 4.3.1–41 apply, but in simpler fashion because polynomial arithmetic is carry-free. Right-to-left division uses 4.7–(3).

Page 624 addendum to answer 4 ———————— 30 Oct 1995

[The numbers a_{np} were first found by Gauss; see his *Werke* **2**, 219–222.]

▸**Page 625** answer 13 ———————— 26 Jun 1997

line 2: $(x^2-\sqrt{-2}x-1)(x^2-\sqrt{-2}x-1) \leadsto (x^2-\sqrt{-2}x-1)(x^2+\sqrt{-2}x-1)$
line 3: $(x^2-\sqrt{-1})(x^2-\sqrt{-1}) \leadsto (x^2-\sqrt{-1})(x^2+\sqrt{-1})$

Page 626 last line of answer 15 ———————— 20 Apr 1996

344–346. $\leadsto$ 344–346. See also H. C. Pocklington, *Proc. Camb. Phil. Soc.* **19** (1917), 57–59.

Page 627 answer 20 ——— 24 Apr 1995

line 2: $u_0/M(u)u_n \rightsquigarrow |u_0|/M(u)$

lines 3–11: affect $|u| \ldots$ (c) $\rightsquigarrow$ affect $\|u\|$, but it changes $|u_0|$ to $M(u)$. (c)

replacement for lines 17–20: (d) $|v_j| \le \binom{m-1}{j}M(v) + \binom{m-1}{j-1}|v_m| \le \binom{m-1}{j}M(u) + \binom{m-1}{j-1}|u_n|$ since $M(v) \le M(u)$ and $|v_m| \le |u_n|$. [M. Mignotte, *Math. Comp.* **28** (1974), 1153–1157.]

Notes: This solution shows that $\binom{m-1}{j}M(u) + \binom{m-1}{j-1}|u_n|$ is an upper bound, so we would like to have a better estimate of $M(u)$. Several methods are known [W. Specht, *Math. Zeit.* **53** (1950), 357–363; Cerlienco, Mignotte, and Piras, *J. Symbolic Comp.* **4** (1987), 21–33]. The simplest and most rapidly convergent is perhaps the following procedure [cf. C. H. Graeffe, *Auflösung der höheren numerischen Gleichungen* (Zürich, 1837)]: Assuming that $u(x) = u_n(x-\alpha_1)\ldots(x-\alpha_n)$, let $\hat u(x) = u(\sqrt{x}\,)u(-\sqrt{x}\,) = (-1)^n u_n^2(x-\alpha_1^2)\ldots(x-\alpha_n^2)$. Then $M(u)^2 = M(\hat u) \le \|\hat u\|$. Hence we may set $c \leftarrow \|u\|$, $v \leftarrow u/c$, $t \leftarrow 0$, and then repeatedly set $t \leftarrow t+1$, $c \leftarrow \|\hat v\|^{1/2^t}c$, $v \leftarrow \hat v/\|\hat v\|$. The invariant relations $M(u) = cM(v)^{1/2^t}$ and $\|v\| = 1$ guarantee that $M(u) \le c$ at each step of the iteration. Notice that when $v(x) = v_0(x^2) + xv_1(x^2)$, we have $\hat v(x) = v_0(x)^2 - xv_1(x)^2$. It can be shown that if each $|\alpha_j|$ is $\le \rho$ or $\ge 1/\rho$, then $M(u) = \|u\|\bigl(1+O(\rho)\bigr)$; hence c will be $M(u)\bigl(1+O(\rho^{2^t})\bigr)$ after t steps.

For example, if $u(x)$ is the polynomial of (22), the successive values of c for $t = 0$, 1, 2, … turn out to be 10.63, 12.42, 6.85, 6.64, 6.65, 6.6228, 6.62246, 6.62246, … . In this example $\rho \approx .90982$. Notice that convergence is not monotonic. Eventually $v(x)$ will converge to the monomial x^m, where m is the number of roots with $|\alpha_j| < 1$, assuming that $|\alpha_j| \ne 1$ for all j; in general, if there are k roots with $|\alpha_j| = 1$, the coefficients of x^m and x^{m+k} will not approach zero, while the coefficients of higher and lower powers of x will.

A famous formula due to Jensen [*Acta Math.* **22** (1899), 359–364] proves that $M(u)$ is the geometric mean of $|u(x)|$ on the unit circle, namely $\exp\bigl(\frac{1}{2\pi}\int_0^{2\pi}\ln|f(e^{i\theta})|\,d\theta\bigr)$. Exercise 21(a) will show, similarly, that $\|u\|$ is the root-mean-square of $|u(x)|$ on the unit circle. The inequality $M(u) \le \|u\|$, which goes back to E. Landau [*Bull. Soc. Math. de France* **33** (1905), 251–261], can therefore be understood as a relation between mean values. Incidentally, Jensen also proved that $\frac{1}{2\pi}\int_0^{2\pi} e^{im\theta}\ln|f(e^{i\theta})|\,d\theta = -\sum_{j=1}^n \alpha_j^m/\bigl(2m\max(|\alpha_j|,1)^{2m}\bigr)$ when $m > 0$. The number $M(u)$ is often called the *Mahler measure* of a polynomial, because Kurt Mahler used it in *Mathematika* **7** (1960), 98–100.

Page 627 replacement for answer 21 ——— 29 Aug 1996

21. (a) The coefficient of $a_{\mathbf p}b_{\mathbf q}c_{\mathbf r}d_{\mathbf s}$ is zero on both sides unless $\mathbf p+\mathbf s = \mathbf q+\mathbf r$. And when this condition holds, the coefficient on the right is $(\mathbf p+\mathbf s)!$; on the left it is

$$\sum_{\mathbf j}\binom{\mathbf p}{\mathbf j}\binom{\mathbf s}{\mathbf r-\mathbf j}\mathbf q!\,\mathbf r! = \binom{\mathbf p+\mathbf s}{\mathbf r}\mathbf q!\,\mathbf r! = (\mathbf q+\mathbf r)!\,.$$

[B. Beauzamy and J. Dégot, *Trans. Amer. Math. Soc.* **345** (1995), 2607–2619; D. Zeilberger, *AMM* **101** (1994), 894–896.]

(b) Let $a_{\mathbf p} = v_{\mathbf p}$, $b_{\mathbf q} = w_{\mathbf q}$, $c_{\mathbf r} = \overline{v}_{\mathbf r}$, $d_{\mathbf s} = \overline{w}_{\mathbf s}$. Then the right side of (a) is $B(u)$, and the left side is a sum of nonnegative terms for each $\mathbf j$ and $\mathbf k$. If we consider only the terms where $\Sigma\,\mathbf j$ is the degree of v, the terms $v_{\mathbf p}/(\mathbf p-\mathbf j)!$ vanish except when $\mathbf p = \mathbf j$.

Those terms therefore reduce to

$$\sum_{\mathbf{j},\mathbf{k}} \frac{1}{\mathbf{j}!\,\mathbf{k}!}\,\bigl|v_{\mathbf{j}}w_{\mathbf{k}}\,\mathbf{j}!\,\mathbf{k}!\bigr|^2 = B(v)B(w)\,.$$

[B. Beauzamy, E. Bombieri, P. Enflo, and H. Montgomery, *J. Number Theory* **36** (1990), 219–245.]

(c) Adding a new variable, if needed to make everything homogeneous, does not change the relation $u = vw$. Thus if v and w have total degrees m and n, respectively, we have $(m+n)!\,[u]^2 \ge m!\,[v]^2\,n!\,[w]^2$; in other words, $[v][w] \le \binom{m+n}{m}^{1/2}[u]$.

Incidentally, one nice way to think of the Bombieri norm is to imagine that the variables are noncommutative. For example, instead of $3xy^3 - z^2w^2$ we could write $\frac34 xyyy + \frac34 yxyy + \frac34 yyxy + \frac34 yyyx - \frac16 zzww - \frac16 zwzw - \frac16 zwwz - \frac16 wzzw - \frac16 wzwz - \frac16 wwzz$. Then the Bombieri norm is the $\|\;\|$ norm on the new coefficients. Another interesting formula, when u is homogeneous of degree n, is

$$[u]^2 = \frac{1}{n!\,\pi^n}\int_{\mathbf{x}}\int_{\mathbf{y}} e^{-x_1^2-\cdots-x_t^2-y_1^2-\cdots-y_t^2}\,\bigl|u(\mathbf{x}+i\mathbf{y})\bigr|^2\,d\mathbf{x}\,d\mathbf{y}\,.$$

(d) The one-variable case corresponds to $t = 2$. Suppose $u = vw$ where v is homogeneous of degree m in t variables. Then $|v_{\mathbf{k}}|^2\,\mathbf{k}!/m! \le [v]^2$ for all $\mathbf{k}$ with $\Sigma\,\mathbf{k} = m$, and $\mathbf{k}! \ge (m/t)!^t$ since $\log\Gamma(x)$ is convex for $x > 0$; therefore $|v_{\mathbf{k}}|^2 \le m!\,[v]^2/(m/t)!^t$. We can assume that $m!\,[v]^2/(m/t)!^t \le m'!\,[w]^2/(m'/t)!^t$, where $m' = n - m$ is the degree of w. Then

$$\begin{aligned}|v_{\mathbf{k}}|^2 \le m!\,[v]^2/(m/t)!^t &\le m!^{1/2}m'!^{1/2}\,[v][w]/(m/t)!^{t/2}(m'/t)!^{t/2}\\ &\le n!^{1/2}\,[u]/(n/2t)!^t\,.\end{aligned}$$

(A better bound is obtained if we maximize the next-to-last expression over all degrees m for which a factor has not been ruled out.) The quantity $n!^{1/4}/(n/2t)!^{t/2}$ is $c_t(2t)^{n/4}n^{-(2t-1)/8}\bigl(1+O(\frac1n)\bigr)$, where $c_t = 2^{1/8}\pi^{-(2t-1)/8}t^{t/4}$ is ≈ 1.004 when $t = 2$.

Notice that we have not demonstrated the existence of an *irreducible* factor with such small coefficients; see exercise 41.

(e) $[u]^2 = \sum_k \binom{n}{k}^2/\binom{2n}{2k} = \sum_k \binom{2k}{k}\binom{2n-2k}{n-k}/\binom{2n}{n} = 4^n/\binom{2n}{n} = \sqrt{\pi n} + O(n^{-1/2})$. Notice that if $v(x) = (x-1)^n$ and $w(x) = (x+1)^n$, we have $[v] = [w] = 2^n$; hence the inequality of (c) is an equality in this case.

(f) Let u and v be homogeneous of degree m and n. Then

$$[uv]^2 \le \sum_{\mathbf{k}} \frac{\bigl(\sum_{\mathbf{j}}|u_{\mathbf{j}}v_{\mathbf{k}-\mathbf{j}}|\bigr)^2}{\binom{m+n}{\mathbf{k}}} \le \sum_{\mathbf{k}}\Biggl(\sum_{\mathbf{j}}\frac{|u_{\mathbf{j}}|^2}{\binom{m}{\mathbf{j}}}\frac{|v_{\mathbf{k}-\mathbf{j}}|^2}{\binom{n}{\mathbf{k}-\mathbf{j}}}\Biggr)\Biggl(\sum_{\mathbf{j}}\frac{\binom{m}{\mathbf{j}}\binom{n}{\mathbf{k}-\mathbf{j}}}{\binom{m+n}{\mathbf{k}}}\Biggr) = [u]^2[v]^2$$

by Cauchy's inequality. [B. Beauzamy, *J. Symbolic Comp.* **13** (1992), 465–472, Proposition 5.]

(g) By exercise 20, $\binom{n}{\lfloor n/2\rfloor}^{-1}M(u)^2 \le \binom{n}{\lfloor n/2\rfloor}^{-1}\|u\|^2 = \binom{n}{\lfloor n/2\rfloor}^{-1}\sum_j |u_j|^2 \le [u]^2 = \sum_j \binom{n}{j}^{-1}|u_j|^2 \le \sum_j \binom{n}{j}M(u)^2 = 2^n M(u)^2$. The upper inequality also follows from (f), for if $u(x) = u_n\prod_{j=1}^n(x-\alpha_j)$ we have $[u]^2 \le |u_n|^2\prod_{j=1}^n[x-\alpha_j]^2 = |u_n|^2\prod_{j=1}^n(1+|\alpha_j|^2) \le |u_n|^2\prod_{j=1}^n\bigl(2\max(1,|\alpha_j|)^2\bigr) = 2^n M(u)^2$.

▸**Page 628** in answer 22 ______ 18 Jun 1995

new paragraph to follow line 10: If p divides q, so that $r = p$, our choices of $V(x)$ and $W(x)$ also satisfy $a(x)V(x) + b(x)W(x) \equiv 1$ (modulo p), as required by Hensel's Lemma.

line 21: In the above notation, $\rightsquigarrow$ When $r = p$ in the notation above,
lines 22 and 24: (modulo pr) $\rightsquigarrow$ (modulo p^2) (3 changes)
line 23: (modulo r) $\rightsquigarrow$ (modulo p)

Page 628 lines -15 through -13 ———— 27 Oct 1995

Hensel's lemma ... generalized $\rightsquigarrow$ "Hensel's Lemma" was actually invented by C. F. Gauss about 1799, in the draft of an unfinished book called *Analysis Residuorum*, §373–374. Gauss incorporated most of the material from that manuscript into his *Disquisitiones Arithmeticæ* (1801), but his ideas about polynomial factorization were not published until after his death [see his *Werke* **2** (Göttingen, 1876), 238]. Hensel's name was attached to the method because it is basic to the theory of p-adic numbers (see exercise 4.1–31). The lemma can be generalized

Page 629 replacement for last line of answer 28 ———— 06 Jun 1995

$O(n^{-2}\log n)$. On the other hand the answers for $1 \le n \le 10$ when $p = 2$ are smaller: 1, $\frac{1}{4}$, $\frac{1}{2}$, $\frac{7}{16}$, $\frac{7}{16}$, $\frac{7}{16}$, $\frac{27}{64}$, $\frac{111}{256}$, $\frac{109}{256}$, $\frac{109}{256}$. A. Knopfmacher and R. Warlimont [*Trans. Amer. Math. Soc.* **347** (1995), 2235–2243] have shown that for fixed p the probability is $c_p + O(1/n)$, where $c_p = \prod_{m\ge 1} e^{-1/m}(1 + a_{mp}/p^m)$ and $c_2 \approx .397$.]

Page 630 line 1 of answer 30 ———— 22 Oct 1995

and $v(x)$ $\rightsquigarrow$ be the "trace" of x and let $v(x)$

▸**Page 630** line 10 of answer 30 ———— 05 Nov 1981

$\bigl(u(x) + u(x)^p\bigr) \bmod g(x)$ $\rightsquigarrow$ $\bigl(t(x) + u(x)^p\bigr) \bmod g(x)$

Page 630 new paragraphs just before answer 31 ———— 14 Sep 1996

The technique of calculating $\gcd\bigl(f(x), T(x) - s\bigr)$ to find factors of $f(x)$ can be traced to A. Arwin, *Arkiv för Matematik, Astronomi och Fysik* **14**, 7 (1918), 1–46; but his method was incomplete because he did not consider $T\bigl(t(x)\bigr)$ for $t(x) \ne x$. A complete factorization algorithm using traces was devised later by R. J. McEliece, *Math. Comp.* **23** (1969), 861–867; see also von zur Gathen and Shoup, *Computational Complexity* **2** (1992), 187–224, Algorithm 3.6.

Henri Cohen has observed that for $p = 2$ it suffices to test at most d special cases $t(x) = x$, x^3, ..., x^{2d-1} when applying this method. One of these choices of $t(x)$ is guaranteed to split $g_d(x)$ whenever g_d is reducible, because we can obtain the effects of all polynomials $t(x)$ of degree $< 2d$ from these special cases using the facts that $T\bigl(t(x)^p\bigr) \equiv T\bigl(t(x)\bigr)$ and $T\bigl(u(x) + t(x)\bigr) \equiv T\bigl(u(x)\bigr) + T\bigl(t(x)\bigr)$ (modulo $g_d(x)$). [*A Course in Computational Algebraic Number Theory* (Springer, 1993), Algorithm 3.4.8.]

▸**Page 631** answer 33 ———— 26 Jun 1997

$p \ge \deg(u)$ $\rightsquigarrow$ $p > \deg(u)$

▸**Page 631** line -16 ———— 14 Sep 1996

[Exercise ... but $\rightsquigarrow$ [Although most polynomials are squarefree, according to exercise 2(b),

▸**Page 631** last two lines of answer 34 ———— 30 Sep 1983

efficiency when the given polynomial is already squarefree. $\rightsquigarrow$ efficiency.

▸**Page 632** line 4 of answer 37 ———— 26 Jun 1997

have certain $\rightsquigarrow$ have a certain

Page 632 new answers ——— 02 Aug 1981

38. (Partial solution by Peter Weinberger.) The average $\leadsto$

38. The conditions imply that when $|z| = 1$ we have either $|u_{n-2}z^{n-2} + \cdots + u_0| < |u_{n-1}| - 1 \le |z^n + u_{n-1}z^{n-1}|$ or $|u_{n-3}z^{n-3} + \cdots + u_0| < u_{n-2} - 1 \le |z^n + u_{n-2}z^{n-2}|$. Therefore by Rouché's theorem, $u(z)$ has at least $n-1$ or $n-2$ roots inside the circle $|z| = 1$. If $u(z)$ is reducible, it can be written $v(z)w(z)$ where v and w are monic integer polynomials. The products of the roots of v and of w are nonzero integers, so each factor has a root of absolute value ≥ 1. Hence the only possibility is that v and w both have exactly one such root and that $u_{n-1} = 0$. These roots must be real, since the complex conjugates are roots; hence $u(z)$ has a real root z_0 with $|z_0| \ge 1$. But this cannot be, for if $r = 1/z_0$ we have $0 = |1+u_{n-2}r^2+\cdots+u_0r^n| \ge 1+u_{n-2}r^2-|u_{n-3}|r^3-\cdots-|u_0|r^n > 1$. [O. Perron, *Crelle* **132** (1907), 288–307; for generalizations, see A. Brauer, *Amer. J. Math.* **70** (1948), 423–432, **73** (1951), 717–720.]

39. First we prove the hint: Let $u(x) = a(x-\alpha_1)\ldots(x-\alpha_n)$ have integer coefficients. The resultant of $u(x)$ with the polynomial $y-t(x)$ is a determinant, so it is a polynomial $r_t(y) = a^{\deg(t)}(y-t(\alpha_1))\ldots(y-t(\alpha_n))$ with integer coefficients (see exercise 4.6.1–12). If $u(x)$ divides $v(t(x))$ then $v(t(\alpha_1)) = 0$, hence $r_t(y)$ has a factor in common with $v(y)$. So if v is irreducible, we have $\deg(u) = \deg(r_t) \ge \deg(v)$.

Given an irreducible polynomial $u(x)$ for which a short proof of irreducibility is desired, we may assume that $u(x)$ is monic, by exercise 18, and that $\deg(u) \ge 3$. The idea is to show the existence of a polynomial $t(x)$ such that $v(y) = r_t(y)$ is irreducible by the criterion of exercise 38. Then all factors of $u(x)$ divide the polynomial $v(t(x))$, and this will prove that $u(x)$ is irreducible. The proof will be succinct if the coefficients of $t(x)$ are suitably small.

The polynomial $v(y) = (y-\beta_1)\ldots(y-\beta_n)$ can be shown to satisfy the criterion of exercise 38 if $n \ge 3$ and $\beta_1\ldots\beta_n \ne 0$, and if the following "smallness condition" holds: $|\beta_j| \le 1/(4n)$ except when $j = n$ or when $\beta_j = \overline{\beta}_n$ and $|\Re\beta_j| \le 1/(4n)$. The calculations are straightforward, using the fact that $|v_0| + \cdots + |v_n| \le (1+|\beta_1|)\ldots(1+|\beta_n|)$.

Let $\alpha_1, \ldots, \alpha_r$ be real and $\alpha_{r+1}, \ldots, \alpha_{r+s}$ be complex, where $n = r + 2s$ and $\alpha_{r+s+j} = \overline{\alpha}_{r+j}$ for $1 \le j \le s$. Consider the linear expressions $S_j(a_0, \ldots, a_{n-1})$ defined to be $\Re\bigl(\sum_{i=0}^{n-1} a_i\alpha_j^i\bigr)$ for $1 \le j \le r+s$ and $\Im\bigl(\sum_{i=0}^{n-1} a_i\alpha_j^i\bigr)$ for $r+s < j \le n$. If $0 \le a_i < b$ and $B = \lceil \max_{j=1}^{n-1}\sum_{i=0}^{n-1}|\alpha_i|^j \rceil$, we have $|S_j(a_1, \ldots, a_{n-1})| < bB$. Thus if we choose $b > (16nB)^{n-1}$, there must be distinct vectors $(a_0, \ldots, a_{n-1})$ and $(a'_0, \ldots, a'_{n-1})$ such that $\lfloor 8nS_j(a_0, \ldots, a_{n-1})\rfloor = \lfloor 8nS_j(a'_0, \ldots, a'_{n-1})\rfloor$ for $1 \le j < n$, since there are b^n vectors but at most $(16nbB)^{n-1} < b^n$ possible $(n-1)$-tuples of values. Let $t(x) = (a_0-a'_0)+\cdots+(a_{n-1}-a'_{n-1})x^{n-1}$ and $\beta_j = t(\alpha_j)$. Then the smallness condition is satisfied. Furthermore $\beta_j \ne 0$; otherwise $t(x)$ would divide $u(x)$. [Cf. *J. Algorithms* **2** (1981), 385–392.]

A completely different approach to this problem has been suggested by Peter Weinberger. He notes first that the average

line 4 of old answer 38: previous answer $\leadsto$ answer to exercise 37
line 6 page 633: "short" proof $\leadsto$ polynomial-time computation

Page 632 line -6 ——— 31 Oct 1995

1973 $\leadsto$ 1972

Page 633 new answer ——— 29 Aug 1996

40. Given a candidate factor $v(x) = x^d + a_{d-1}x^{d-1} + \cdots + a_0$, change each a_j to a rational fraction (modulo p^e), with numerators and denominators $\le B$. Then multiply by the least common denominator, and see if the resulting polynomial divides $u(x)$ over the integers. If not, no factor of $u(x)$ with coefficients bounded by B is congruent modulo p^e to a multiple of $v(x)$.

Page 633 new answer ——— 05 Oct 1996

41. David Boyd notes that $4x^8 + 4x^6 + x^4 + 4x^2 + 4 = (2x^4 + 4x^3 + 5x^2 + 4x + 2) \times (2x^4 - 4x^3 + 5x^2 - 4x + 2)$, and he has found examples of higher degree to prove that c must be > 2 if it exists.

▸**Page 634** line 1 of answer 5 ——— 28 Jun 1997

$1 \le j \le 2^r \leadsto 0 \le j \le 2^r$

Page 635 new answer ——— 23 Oct 1995

15. Achim Flammenkamp [Diplomarbeit in Mathematics (Bielefeld University, 1991), Part 1] has shown that the numbers n with $\lambda(n) + 3 = l(n) < l^*(n)$ all have the form $2^A + 2^B + 2^C + 2^D + 2^E$ where $A > B > C > D > E$ and $B + E = C + D$; moreover, they are described precisely by the following eight patterns where $|\epsilon| \le 1$: $2^A + 2^{A-3} + 2^C + 2^{C-1} + 2^{2C+2-A}$, $2^A + 2^{A-1} + 2^C + 2^D + 2^{C+D+1-A}$, $2^A + 2^B + 2^{2B-A+3} + 2^{2B+2-A} + 2^{3B+5-2A}$, $2^A + 2^B + 2^{2B-A+\epsilon} + 2^D + 2^{B+D+\epsilon-A}$, $2^A + 2^B + 2^{B-1} + 2^D + 2^{D-1}$, $2^A + 2^B + 2^{B-2} + 2^D + 2^{D-2}$ $(A > B + 1)$, $2^A + 2^B + 2^C + 2^{2B+\epsilon-A} + 2^{B+C+\epsilon-A}$, $2^A + 2^B + 2^C + 2^{B+C+\epsilon-A} + 2^{2C+\epsilon-A}$.

Page 637 end of answer 26 ——— 21 May 1981

S. Brown, ... 190.] $\leadsto$ Spencer Brown, *Comp. J.* **9** (1966), 188–190.] In fact, as Richard Brent has observed, the number of operations can be reduced to $O(d^2 \log n)$, or even to $O(d \log d \log n)$ using exercise 4.7–6, if we first compute $x^n \bmod (x^d - a_1 x^{d-1} - \cdots - a_d)$ and then replace x^j by x_j.

Page 638 replacement for lines 5–8 of answer 30 ——— 24 Aug 1996

$\overline{\nu}(n)$ as defined in exercise 4.1–34. A generalized right-to-left binary method for exponentiation, which uses $\lambda(n) + \overline{\nu}(n) - 1$ multiplications when both x and x^{-1} are given, can be based on the representation α_n of that exercise.]

▸**Page 638** answer 33 ——— 28 Jun 1997

line 7: exercise 4.5.4–35 $\leadsto$ exercise 4.5.4–36
line 8: $(1 + \frac{1}{2}\ln 2) \leadsto \frac{1}{2}\ln 2$

Page 638 answer 35 ——— 18 Oct 1996

119. $\leadsto$ 119. [H. Zantema has analyzed the analogous problem when the cost of step i is $a_j + a_k$ instead of $a_j a_k$; see *J. Algorithms* **12** (1991), 281–307. In this case the optimum chains have total cost $\frac{5}{2}n + O(n^{1/2})$. Furthermore the optimum additive cost when n is odd is at least $\frac{5}{2}(n-1)$, with equality if and only if n can be written as a product of numbers of the form $2^k + 1$.]

Page 639 line 8 of answer 39 ——— 15 Jun 1981

arcs. $\leadsto$ arcs. [Cf. *J. Algorithms* **2** (1981), 13–21.]

Page 639 answer 41 ——— 01 Oct 1982

line 1: To appear. $\leadsto$ *SICOMP* **10** (1981), 638–646.
line 3: NP hard $\leadsto$ NP-hard
lines 3 and 4: NP complete $\leadsto$ NP-complete

▸**Page 640** answer 7 —— 26 Feb 1996

line 2: for $j = k,\ k+1,\ \ldots,\ n$ $\leadsto$ for $j = n,\ n-1,\ \ldots,\ k$
append to line 3: Beware of rounding errors.

Page 641 new answer —— 27 Apr 1995

11. Efficient methods are known for computing an approximate value, if the matrix is sufficiently dense; see A. Sinclair, *Algorithms for Random Generation and Counting* (Boston: Birkhäuser, 1993). But this problem asks for the exact value. There may be a way to evaluate the permanent with $O(c^n)$ operations for some $c < 2$.

Page 641 lines 20–23 —— 31 Jul 1996

The best ... 341]. $\leadsto$ For $n = 5$, the current record is 100 noncommutative multiplications [O. M. Makarov, *USSR Comp. Math. and Math. Phys.* **27**, 1 (1987), 205–207]. The best lower bounds known so far are due to J.-C. Lafon and S. Winograd, who showed that $2n^2 - 1$ nonscalar multiplications are necessary, and $mn + ns + m - n - 1$ in the $m \times n \times s$ case ["A lower bound for the multiplicative complexity of the product of two matrices," Centre de Calcul, Univ. Louis Pasteur (Strasbourg, 1979)]. If all calculations must be done over the integers, slightly better lower bounds were obtained by N. H. Bshouty [*SICOMP* **18** (1989), 759–765], who proved that $m \times n$ by $n \times s$ matrix multiplication mod 2 requires at least $\sum_{k=0}^{j-1} \lfloor ms/2^k \rfloor + \frac{1}{2}(n + (n \bmod j)) \times (n - (n \bmod j) - j) + n \bmod j$ multiplications when $n \ge s \ge j \ge 1$; setting $m = n = s$ and $j \approx \lg n$ gives $2.5n^2 - \frac{1}{2} n \lg n + O(n)$.

Page 642 lines 9–15 —— 08 Jul 1996

is essentially due ... generalized by J. W. Cooley ... 1677. Details ... 78–83. $\leadsto$ was discovered by C. F. Gauss in 1805 and independently rediscovered many times since, most notably by J. W. Cooley ... 1677; M. T. Heideman, D. H. Johnson, and C. S. Burrus, *IEEE ASSP Magazine* **1**, 4 (Oct. 1984), 14–21. Details concerning its use have been discussed by hundreds of authors, admirably summarized by Charles Van Loan, *Computational Frameworks for the Fast Fourier Transform* (Philadelphia: SIAM, 1992).

Page 642 end of answer 17 —— 26 Oct 1995

1972)]. $\leadsto$ 1972)]. C. Pandu Rangan showed that if we compute the polynomial as $L_1 R_1 + \cdots + L_{n-1} R_{n-1}$, where the L's and R's are linear combinations of the x's, at least $n - 2$ additions are needed to form the L's and R's [*J. Algorithms* **4** (1983), 282–285]. But his lower bound does not obviously apply to all polynomial chains.

▸**Page 644** line 6 —— 05 Jul 1997

γ_{2x} $\leadsto$ $\gamma_2 x$

▸**Page 647** lines 11–13 of answer 40 —— 30 Sep 1982

$$v(x) = \bigl((x+\alpha)^2 + \beta\bigr)(x+\gamma), \qquad w(x) = v(x) + x,$$
$$t_1(x) = \bigl(v(x) + \delta_1\bigr)\bigl(w(x) + \epsilon_1\bigr), \qquad t_2(x) = \bigl(v(x) + \delta_2\bigr)\bigl(w(x) + \epsilon_2\bigr),$$
$$u(x) = \bigl(t_1(x) + \zeta\bigr)\bigl(t_2(x) - t_1(x) + \eta\bigr) + \kappa.$$

Page 647 answer 41 —— 01 Jan 1982

line 6: (1960) $\leadsto$ (1971)

line 12: [Helmut Alt and Jan van Leeuwen $\rightsquigarrow$ Helmut Alt and Jan van Leeuwen [*Computing* **27** (1981), 205–215]
lines 14–15: It is unknown ... divisions.] $\rightsquigarrow$ Six multiplication/division operations and three addition/subtractions are necessary and sufficient to compute $(a+bi)/(c+di)$ [T. Lickteig, *SICOMP* **16** (1987), 278–311].

Page 648 line 18 ———— 04 Sep 1996

158. $\rightsquigarrow$ 158; J. Ganz, *SICOMP* **24** (1995), 473–483.

▶**Page 648** line 4 of answer 44 ———— 05 Jul 1997

$(\sum_{1\le r\le s} H_{kr}c_{rl})$ $\rightsquigarrow$ $(\sum_{1\le h\le s} H_{kh}c_{hl})$

▶**Page 648** line 4 of answer 45 ———— 05 Sep 1996

a give $\rightsquigarrow$ a given

Page 651 last line of answer 53 ———— 01 Jul 1996

117.] $\rightsquigarrow$ 117; *IEEE Trans.* **ASSP-27** (1979), 169–181.]

▶**Page 651** line 2 of answer 55 ———— 06 Jul 1997

r' is $\rightsquigarrow$ q' is

▶**Page 652** line 8 of answer 57 ———— 07 Sep 1989

$N < 4n$ $\rightsquigarrow$ $N \le 4n$

▶**Page 652** line 13 of answer 58 ———— 05 Jul 1997

$t \ge 3.52n$ $\rightsquigarrow$ $r \ge 3.52n$

Page 653 line 3 ———— 31 Oct 1995

streamlined. $\rightsquigarrow$ streamlined. For example, the final value of $Z_{2m-1}(w)$ in step N5 will always be zero.

▶**Page 653** answer 59 ———— 24 Jul 1981

Step C2 line 3: x_{2m-1} $\rightsquigarrow$ $x_{2m-1}u^{m-1}$
Step C4 line 2: $(z_0,\ldots,z_{m-1})$ $\rightsquigarrow$ $(z_0,\ldots,z_{2m-1})$
Step N2 line 1: $x_{mj+i}\,Y_{ij}$ $\rightsquigarrow$ $x_{mj+i},\ Y_{ij}$
Step N2 line 4: (u^n+1) $\rightsquigarrow$ $(u^{2^n}+1)$
Step N2 line 6: correlation $\rightsquigarrow$ convolution
Step N2 line 6: $x(u)y(u) =$ $\rightsquigarrow$ $x(u)y(u) \equiv$
Step N3 lines 7–10: $(X_{s+t}(w)\ \ldots$ Do $\rightsquigarrow$ $\bigl(X_{s+t}(w) + w^{(r/m)s'}X_{s+t+2^j}(w),\ X_{s+t}(w) - w^{(r/m)s'}X_{s+t+2^j}(w)\bigr)$, where $s' = 2^j(s_{j+1}\ldots s_{\lfloor n/2\rfloor})_2$. (We are evaluating 4.3.3–(34), with $K = 2m$ and $\omega = w^{r/m}$; notice the bit-reversal in s'.) The operation $X_i(w) \leftarrow X_i(w) + w^k X_l(w)$ means, more precisely, that we set $X_{ij} \leftarrow X_{ij} + X_{l(j-k)}$ for $k \le j < r$, and $X_{ij} \leftarrow X_{ij} - X_{l(j-k+r)}$ for $0 \le j < k$. A copy of $X_l(w)$ can be made without wasting much space.) Do
Step N5 line 2: $w^{-(r/m)(s/2)}$ $\rightsquigarrow$ $w^{-(r/m)s'}$

Page 654 top ———— 10 Jan 1997

line 2: i.e., $\rightsquigarrow$ for example,
line 4: throughout. $\rightsquigarrow$ throughout. All of the z and Z variables will be bounded by $(2^nM)^2$, which is n more bits than required to hold the final convolution.

▸**Page 654** near the top — 26 Mar 1982

line 7: $A_{\lfloor n/2 \rfloor} \leadsto A_{\lceil n/2 \rceil}$
line 7: $D_{\lfloor n/2 \rfloor} \leadsto D_{\lceil n/2 \rceil}$
line 8: $M_{\lfloor n/2 \rfloor} \leadsto M_{\lceil n/2 \rceil}$

Page 654 line 11 — 31 Oct 1995

$S_n \le \frac{1}{2}n \lg n + n.$ $\leadsto$ $S_n \le S_{n+1} \le \frac{1}{2}n \lg n + n$ for all $n \ge 1$.
[Also delete lines 13–17.]

▸**Page 654** answer 61 — 02 Jul 1997

line 1: $a_{ij}(u) \leadsto a_{il}(u)$
line 1: $a_{il\mu}u^\mu \leadsto \sum_\mu a_{il\mu}u^\mu$
line 4: $\sum_{\mu+\nu+\sigma} \leadsto \sum_{\mu+\nu+\sigma=d}$

Page 655 new answer — 26 Oct 1995

65. Let $l = \lfloor \lg n \rfloor$, and suppose x, x^2, x^4, $\ldots$, x^{2^l} have been precomputed. If $u(x)$ is monic of degree $n = 2m+1$, we can write $u(x) = (x^{m+1} + \alpha)v(x) + w(x)$, where $v(x)$ and $w(x)$ are monic of degree m. This yields a method for $n = 2^{l+1} - 1 \ge 3$ that requires $2^l - 1$ further multiplications and $2^{l+1} + 2^{l-1} - 2$ additions. If $n = 2^l$ we can apply Horner's rule to reduce n by 1. And if $m = 2^l < n < 2^{l+1} - 1$, we can write $u(x) = x^m v(x) + w(x)$ where v and w are monic of degrees $n-m$ and m, respectively; by induction on l, this requires at most $\frac{1}{2}n + l - 1$ multiplications and $\frac{5}{4}n$ additions, after the precomputation. [Cf. S. Winograd, *IBM Tech. Disclosure Bull.* **13** (1970), 1133–1135.]

Note: It is also possible to evaluate $u(x)$ with $\frac{1}{2}n + O(\sqrt{n}\,)$ multiplications and $n + O(\sqrt{n}\,)$ additions, under the same ground rules, if our goal is to minimize multiplications + additions. The generic polynomial

$$p_{jkm}(x) = \biggl(\Bigl(\ldots(((x^m + \alpha_0)(x^{j+1} + \beta_1) + \alpha_1)(x^{j+2} + \beta_2) \\ + \alpha_2)\ \cdots\Bigr)(x^k + \beta_{k-j}) + \alpha_{k-j}\biggr)(x^j + \beta_0)$$

"covers" the coefficients of exponents $\{j, j+k, j+k+(k-1), \ldots, j+k+(k-1)+\cdots+(j+1), m'-k, m'-k+1, \ldots, m'-j\}$, where $m' = m + j + (j+1) + \cdots + k = m + \binom{k+1}{2} - \binom{j}{2}$. By adding together such polynomials $p_{1km_1}(x), p_{2km_2}(x), \ldots, p_{kkm_k}(x)$ for $m_j = \binom{j+1}{2} + \binom{k-j+2}{2}$, we obtain an arbitrary monic polynomial of degree k^2+k+1. [Rabin and Winograd, *Comm. on Pure and Applied Math.* **25** (1972), 433–458, §2; this paper also proves that constructions with $\frac{1}{2}n + O(\log n)$ multiplications and $\le (1+\epsilon)n$ additions are possible for all $\epsilon > 0$, if n is large enough.]

Page 655 new answer — 04 Sep 1996

66. Let $y_{ij} = x_{ij} - [i=j]$, and apply the recursive construction (31) to the matrix $I + Y$, using arithmetic on power series in the n^2 variables y_{ij} but ignoring all terms of total degree $> n$. Each entry h of the array is represented as a sum $h_0 + h_1 + \cdots + h_n$, where h_k is the value of a homogeneous polynomial of degree k. Then every addition step becomes $n+1$ additions, and every multiplication step becomes $\approx \frac{1}{2}n^2$ multiplications and $\approx \frac{1}{2}n^2$ additions. Furthermore, every division is by a quantity of the form $1 + h_1 + \cdots + h_n$, since all divisions in the recursive construction are by 1 when the y_{ij} are entirely zero; therefore division is slightly easier than multiplication (see Eq. 4.7–(3) when $V_0 = 1$). Since we stop when reaching a 2×2 determinant, we need not subtract 1 from y_{jj} when $j > n-2$. It turns out that when redundant computations are suppressed, this method requires $20\binom{n}{5}+8\binom{n}{4}+12\binom{n}{3}-4\binom{n}{2}+5n-4$ multiplications and $20\binom{n}{5}+8\binom{n}{4}+4\binom{n}{3}+24\binom{n}{2}-n$ additions, thus $\frac{1}{6}n^5-O(n^4)$ of each. A similar method can be used to eliminate division in many other cases; see *Crelle* **264** (1973), 184–202. (But the next exercise constructs an even faster divisionless scheme for determinants.)

Page 655 new answer ________________ 02 Oct 1996

67. Set $A = \lambda - x$, $B = -u$, $C = -v$, and $D = \lambda I - Y$ in the hinted identity, then take the determinant of both sides, using the fact that $I/\lambda + Y/\lambda^2 + Y^2/\lambda^3 + \cdots$ is the inverse of D as a formal power series in $1/\lambda$. We need to compute $uY^k v$ only for $0 \le k \le n-2$, because we know that $f_X(\lambda)$ is a polynomial of degree n; thus, only $n^3+O(n^2)$ multiplications and $n^3+O(n^2)$ additions are needed to advance from degree $n-1$ to degree n. Proceeding recursively, we obtain the coefficients of f_X from the elements of X after doing $6\binom{n}{4}+7\binom{n}{3}+2\binom{n}{2}$ multiplications and $6\binom{n}{4}+5\binom{n}{3}+2\binom{n}{2}$ addition/subtractions.

If we only want to compute $\det X = (-1)^n f_X(0)$, we save $3\binom{n}{2}-n+1$ multiplications and $\binom{n}{2}$ additions. This division-free method for determinant evaluation is in fact quite economical when n has a moderate size; it beats the obvious cofactor expansion scheme when $n > 4$.

If matrix multiplication can be done in $O(n^\beta)$ steps, the same approach leads to a division-free computation in $O(n^{\beta+1}\log n)$ steps, because the vectors uY^k for $0 \le k < n$ can be evaluated in $O(n^\beta \log n)$ steps: Take a matrix whose first 2^l rows are uY^k for $0 \le k < 2^l$ and multiply it by Y^{2^l}; then the first 2^l rows of the product are uY^k for $2^l \le k < 2^{l+1}$. [Cf. S. J. Berkowitz, *Inf. Processing Letters* **18** (1984), 147–150.] Of course such asymptotically "fast" multiplication is strictly of theoretical interest.

Page 655 new answer ________________ 04 Sep 1996

68. Suppose $g_1 = u_1 \circ v_1$, ..., $g_r = u_r \circ v_r$, and $f = \alpha_1 g_1 + \cdots + \alpha_r g_r + p_0$, where $u_k = \beta_{k1}g_1+\cdots+\beta_{k(k-1)}g_{k-1}+p_k$, $v_k = \gamma_{k1}g_1+\cdots+\gamma_{k(k-1)}g_{k-1}+q_k$, each $\circ$ is "$\times$" or "/", and each p_j or q_j is a polynomial of degree ≤ 1 in x_1, ..., x_n. Compute auxiliary quantities w_k, y_k, z_k for $k = r$, $r-1$, ..., 1 as follows: $w_k = \alpha_k + \beta_{(k+1)k}y_{k+1} + \gamma_{(k+1)k}z_{k+1} + \cdots + \beta_{rk}y_r + \gamma_{rk}z_r$, and

$$\begin{aligned} y_k &= w_k \times v_k, & z_k &= w_k \times u_k, & &\text{if } g_k = u_k \times v_k; \\ y_k &= w_k / v_k, & z_k &= -y_k \times g_k, & &\text{if } g_k = u_k / v_k. \end{aligned}$$

Then $f' = p_0' + p_1' y_r + q_1' z_1 + \cdots + p_r' y_r + q_r' z_r$, where $'$ denotes the derivative with respect to any of x_1, ..., x_n. [W. Baur and V. Strassen, *Theoretical Comp. Sci.* **22** (1983), 317–330.] We save two chain multiplications if $g_r = u_r \times v_r$, since $w_r = \alpha_r$. Repeating the construction gives all second partial derivatives with at most $9m + 3d$ chain multiplications and $4d$ divisions.

Page 655 new answer ________________ 04 Jan 1997

69. (a) If $k = (k_1, \ldots, k_s)^T$ is a vector of relatively prime integers, so is Uk, since any common divisor of the elements of Uk divides all elements of $k = U^{-1}Uk$. Therefore VUk cannot have all integer components.

(b) Suppose there is a polynomial chain for Vx with t multiplications. If $t = 0$, the entries of V must all be integers, so $s = 0$. Otherwise let $\lambda_i = \alpha \times \lambda_k$ or $\lambda_i = \lambda_j \times \lambda_k$ be the first multiplication step. We can assume that $\lambda_k = n_1 x_1 + \cdots + n_s x_s + \beta$ where

n_1, ..., n_s are integers, not all zero, and β is constant. Find a unimodular matrix U such that $(n_1, \ldots, n_s)U = (0, \ldots, 0, d)$, where $d = \gcd(n_1, \ldots, n_s)$. (The algorithm discussed before Eq. 4.5.2–(14) implicitly defines such a U.) Construct a new polynomial chain with inputs y_1, ..., y_{s-1} as follows: First calculate $x = (x_1, \ldots, x_s)^T = U(y_1, \ldots, y_{s-1}, -\beta/d)^T$, then continue with the assumed polynomial chain for Vx. When step i of the latter chain is reached, we will have $\lambda_k = (n_1, \ldots, n_s)x + \beta = 0$, so we can simply set $\lambda_i = 0$ instead of multiplying. After Vx has been evaluated, add the constant vector $w\beta/d$ to the result, where w is the rightmost column of VU, and let W be the other $s-1$ columns of VU. The new polynomial chain has computed $Vx + w\beta/d = VU(y_1, \ldots, y_{s-1}, -\beta/d)^T + w\beta/d = W(y_1, \ldots, y_{s-1})^T$, with $t-1$ multiplications. But the columns of W are Z-independent, by part (a); hence $t-1 \ge s-1$, by induction on s, and we have $t \ge s$.

composition

(c) Let $x_j = 0$ for the $t-s$ values of j that aren't in the set of Z-independent columns. Any chain for Vx then evaluates $V'x'$ for a matrix V' to which part (b) applies.

(d) $\lambda_1 = x - y$, $\lambda_2 = \lambda_1 + \lambda_1$, $\lambda_3 = \lambda_2 + x$, $\lambda_4 = (1/6) \times \lambda_3$, $\lambda_5 = \lambda_4 + \lambda_4$, $\lambda_6 = \lambda_5 + y$ $(= x + y/3)$, $\lambda_7 = \lambda_6 - \lambda_1$, $\lambda_8 = \lambda_7 + \lambda_4$ $(= x/2 + y)$. But $\{x/2+y, x+y/2\}$ needs two multiplications, since the columns of $\left(\begin{smallmatrix}1/2 & 1\\ 1 & 1/2\end{smallmatrix}\right)$ are Z-independent. [*Journal of Information Processing* **1** (1978), 125–129.]

Page 656 answer 4 ——— 29 Apr 1995

line 1: $W_0 = 1 \rightsquigarrow W_0 = e^{V_0}$
replacement for lines 3 and 4: If $W(z) = \ln V(z)$, the roles of V and W are reversed; hence when $V_0 = 1$ the rule is $W_0 = 0$ and $W_n = V_n + \sum_{k=1}^{n-1}(k/n - 1)V_k W_{n-k}$ for $n \ge 1$.

▸**Page 656** line 1 of answer 6 ——— 13 Jul 1997

$\varphi(x) \rightsquigarrow \phi(x)$

Page 656 replacement for answer 8 ——— 03 May 1995

8. G1. Input G_1 and V_1; set $n \leftarrow 1$, $U_0 \leftarrow 1/V_1$; output $W_1 = G_1 U_0$.

G2. Increase n by 1. Terminate the algorithm if $n > N$; otherwise input V_n and G_n.

G3. Set $U_k \leftarrow \bigl(U_k - \sum_{j=1}^{k} U_{k-j}V_{j+1}\bigr)/V_1$ for $k = 0, 1, \ldots, n-2$ (in this order); then set $U_{n-1} \leftarrow -\sum_{k=2}^{n} kU_{n-k}V_k/V_1$.

G4. Output $W_n = \sum_{k=1}^{n} kU_{n-k}G_k/n$ and return to G2. ▌

(The running time of the order N^3 algorithm is hereby increased by only order N^2.)

Note: Algorithms T and N determine $V^{[-1]}\bigl(U(z)\bigr)$; the algorithm in this exercise determines $G\bigl(V^{[-1]}(z)\bigr)$, which is somewhat different. Of course, the results can all be obtained by a sequence of operations of reversion and composition (exercise 11), but it is helpful to have more direct algorithms for each case.

Page 657 line 12 ——— 03 May 1995

$U\bigl((V^{-1})^{-1}(z)\bigr) - U_0$. $\rightsquigarrow$ $U\bigl(V^{[-1][-1]}(z)\bigr) - U_0$. See also exercise 20.

Page 657 last two lines of answer 13 ——— 13 Dec 1980

Fred ... (1979). $\rightsquigarrow$ Richard P. Brent, Fred G. Gustavson, and David Y. Y. Yun, *J. Algorithms* **1** (1980), 259–295.

▸**Page 658** last line (of answer 14) ________ 14 Jul 1997

$V'(z) \leadsto V'(P(z))$

Page 658 new answer ________ 03 May 1995

15. The differential equation $U'(z)/U(z)^k = 1/z^k$ implies that $U(z)^{1-k} = z^{1-k} + c$ for some constant c. So we find $U^{[n]}(z) = z/(1 + cnz^{1-k})^{1/(k-1)}$.

A similar argument solves (27) for arbitrary $V(z)$: If $W'(z) = 1/V(z)$, we have $W(U^{[n]}(z)) = W(z) + nc$ for some c.

Page 658 new answer ________ 01 May 1995

16. We want to show that $[t^n]\, t^{n+1}\big((n+1)R'_{k+1}(t)/V(t)^n - nR'_k(t)/V(t)^{n+1}\big) = 0$. This follows since $(n+1)R'_{k+1}(t)/V(t)^n - nR'_k(t)/V(t)^{n+1} = \frac{d}{dt}\big(R_{k+1}(t)/V(t)^n\big)$. Consequently we have $n^{-1}[t^{n-1}]\, R'_1(t)\, t^n/V(t)^n = (n-1)^{-1}[t^{n-2}]\, R'_2(t)\, t^{n-1}/V(t)^{n-1} = \cdots = 1^{-1}[t^0]\, R'_n(t)\, t/V(t) = [t]\, R_n(t)/V_1 = W_n$.

Page 658 new answers ________ 04 May 1995

17. Equating coefficients of $x^l y^m$, the convolution formula states that $\binom{l+m}{m} v_{n(l+m)} = \sum_k \binom{n}{k} v_{kl} v_{(n-k)m}$, which is the same as $[z^n]\, V(z)^{l+m} = \sum_k \big([z^k]V(z)^l\big)\big([z^{n-k}]\, V(z)^m\big)$, which is a special case of (2).

Notes: The name "poweroid" was introduced by J. F. Steffensen, who was the first of many authors to study the striking properties of these polynomials in general [*Acta Mathematica* **73** (1941), 333–366]. For a review of the literature, and for further discussion of the topics in the next several exercises, see D. E. Knuth, *The Mathematica Journal* **2** (1992), 67–78. One of the results proved in that paper is the asymptotic formula $V_n(x) = e^{xV(s)}\big(\frac{n}{es}\big)^n\big(1 - V_2 y + O(y^2) + O(x^{-1})\big)$, if $V_1 = 1$ and $sV'(s) = y$ and $y = n/x$ is bounded as $x \to \infty$ and $n \to \infty$.

18. We have $V_n(x) = \sum_k x^k n!\, [z^n]\, V(z)^k/k! = n!\, [z^n]\, e^{xV(z)}$. Consequently $V_n(x)/x = (n-1)!\, [z^{n-1}]\, V'(z)\, e^{xV(z)}$ when $n > 0$. We get the stated identity by equating the coefficients of z^{n-1} in $V'(z)\, e^{(x+y)V(z)} = V'(z)\, e^{xV(z)} e^{yV(z)}$.

19. We have

$$v_{nm} = \frac{n!}{m!}\,[z^n]\left(\frac{v_1}{1!}z + \frac{v_2}{2!}z^2 + \frac{v_3}{3!}z^3 + \cdots\right)^m$$

$$= \sum_{\substack{k_1+k_2+\cdots+k_n=m\\ k_1+2k_2+\cdots+nk_n=n\\ k_1,k_2,\ldots,k_n\ge 0}} \frac{n!}{k_1!\,k_2!\ldots k_n!}\left(\frac{v_1}{1!}\right)^{k_1}\left(\frac{v_2}{2!}\right)^{k_2}\cdots\left(\frac{v_n}{n!}\right)^{k_n}$$

by the multinomial theorem 1.2.6–(39). These coefficients arise also in Arbogast's formula, exercise 1.2.5–21, and we can associate the terms with set partitions as explained in the answer to that exercise. The recurrence

$$v_{nk} = \sum_j \binom{n-1}{j-1} v_j\, v_{(n-j)(k-1)}$$

shows how to calculate column k from columns 1 and $k-1$; it is readily interpreted with respect to partitions of $\{1,\ldots,n\}$, since there are $\binom{n-1}{j-1}$ ways to include the element n

in a subset of size j. The first few rows of the matrix are

$$\begin{matrix} v_1 \\ v_2 & v_1^2 \\ v_3 & 3v_1v_2 & v_1^3 \\ v_4 & 4v_1v_3+3v_2^2 & 6v_1^2v_2 & v_1^4 \\ v_5 & 5v_1v_4+10v_2v_3 & 15v_1v_2^2+10v_1^2v_3 & 10v_1^3v_2 & v_1^5 \end{matrix}\quad.$$

20. $[z^n]\,W(z)^k = \sum_j\bigl([z^j]\,U(z)^k\bigr)\bigl([z^n]\,V(z)^j\bigr)$; hence $w_{nk} = (n!/k!)\sum_j\bigl((k!/j!)u_{jk}\bigr)\times\bigl((j!/n!)v_{nj}\bigr)$. [E. Jabotinski, *Comptes Rendus Acad. Sci.* **224** (Paris, 1947), 323–324.]

21. (a) If $U(z) = \alpha W(\beta z)$ we have $u_{nk} = \frac{n!}{k!}\,[z^n]\,\bigl(\alpha W(\beta(z)\bigr)^k = \alpha^k\beta^n w_{nk}$; in particular, if $U(z) = V^{[-1]}(z) = -W(-z)$ we have $u_{nk} = (-1)^{n-k}w_{nk}$. So $\sum_k u_{nk}v_{km}$ and $\sum_k v_{nk}u_{km}$ correspond to the identity function z, by exercise 20.

(b) This identity is, in fact, equivalent to Lagrange's inversion formula: We have $w_{nk} = (-1)^{n-k}u_{nk} = (-1)^{n-k}\frac{n!}{k!}\,[z^n]\,V^{[-1]}(z)^k$, and the coefficient of z^n in $V^{[-1]}(z)^k$ is $n^{-1}\,[t^{n-1}]\,kt^{n+k-1}/V(t)^n$ by exercise 16. On the other hand we have defined $v_{(-k)(-n)}$ to be $(-k)^{\underline{n-k}}\,[z^{n-k}]\,\bigl(V(z)/z)\bigr)^{-n} = (-1)^{n-k}(n-1)\ldots(k+1)k\,[z^{n-1}]\,z^{n+k-1}/V(z)^n$.

22. (a) If $V(z) = U^{\{\alpha\}}(z)$ and $W(z) = V^{\{\beta\}}(z)$, we have $W(z) = V\bigl(zW(z)^\beta\bigr) = U\bigl(zW(z)^\beta\,V\bigl(zW(z)^\beta\bigr)^\alpha\bigr) = U\bigl(zW(z)^{\alpha+\beta}\bigr)$. (Notice the contrast between this law and the similar formulas $U^{[1]}(z) = U(z)$, $U^{[\alpha][\beta]}(z) = U^{[\alpha\beta]}(z)$ that apply to iteration.)

(b) $B^{\{2\}}(z)$ is the generating function for binary trees, 2.3.4.4–(12), which is $W(z)/z$ in the example $z = t - t^2$ following Algorithm L. Moreover, $B^{\{t\}}(z)$ is the generating function for t-ary trees, exercise 2.3.4.4–11.

(c) The hint is equivalent to $zU^{\{\alpha\}}(z)^\alpha = W^{[-1]}(z)$, which is equivalent to the formula $zU^{\{\alpha\}}(z)^\alpha/U\bigl(zU^{\{\alpha\}}(z)^\alpha\bigr)^\alpha = z$. Now Lagrange's inversion theorem (exercise 8) says that $[z^n]\,W^{[-1]}(z)^x = \frac{x}{n}[z^{-x}]\,W(z)^{-n}$ when x is a positive integer. (Here $W(z)^{-n}$ is a Laurent series — a power series divided by a power of z; we can use the notation $[z^m]\,V(z)$ for Laurent series as well as for power series.) Therefore $[z^n]\,U^{\{\alpha\}}(z)^x = [z^n]\,\bigl(W^{[-1]}(z)/z\bigr)^{x/\alpha} = [z^{n+x/\alpha}]\,W^{[-1]}(z)^{x/\alpha}$ is equal to $\frac{x/\alpha}{n+x/\alpha}\,[z^{-x/\alpha}]\,W(z)^{-n-x/\alpha} = \frac{x}{x+n\alpha}\,[z^{-x/\alpha}]\,z^{-n-x/\alpha}U(z)^{x+n\alpha}$ when x/α is a positive integer. We have verified the result for infinitely many α; that is sufficient, since the coefficients of $U^{\{\alpha\}}(z)^x$ are polynomials in α.

We've seen special cases of this result in exercises 1.2.6–25 and 2.3.4.4–29. One memorable consequence of the hint is the case $\alpha = -1$:

$$W(z) = zU(z) \qquad \text{if and only if} \qquad W^{[-1]}(z) = z/U^{\{-1\}}(z)\,.$$

(d) If $U_0 = 1$ and $V_n(x)$ is the poweroid for $V(z) = \ln U(z)$, we've just proved that $xV_n(x+n\alpha)/(x+n\alpha)$ is the poweroid for $\ln U^{\{\alpha\}}(z)$. So we can plug this poweroid into the former identities, changing y to $y - \alpha n$ in the second formula.

23. (a) We have $U = I + T$ where T^n is zero in rows $\le n$. Hence $\ln U = T - \frac{1}{2}T^2 + \frac{1}{3}T^3 - \cdots$ will have the property that $\exp(\alpha\ln U) = I + \binom{\alpha}{1}T + \binom{\alpha}{2}T^2 + \cdots = U^\alpha$. Each entry of U^α is a polynomial in α, and the relations of exercise 19 hold whenever α is a positive integer; therefore U^α is a power matrix for all α, and its first column defines $U^{[\alpha]}(z)$. (In particular, U^{-1} is a power matrix; this is another way to revert $U(z)$.)

(b) Since $U^\epsilon = I + \epsilon\ln U + O(\epsilon^2)$, we have $l_{nk} = [\epsilon]u_{nk}^{[\epsilon]} = \frac{n!}{k!}\,[z^n][\epsilon]\,\bigl(z + \epsilon L(z) + O(\epsilon^2)\bigr)^k = \frac{n!}{k!}\,[z^n]\,kz^{k-1}L(z)$.

(c) $\frac{\partial}{\partial\alpha}U^{[\alpha]}(z) = [\epsilon]\,U^{[\alpha+\epsilon]}(z)$, and we have $U^{[\alpha+\epsilon]}(z) = U^{[\alpha]}\bigl(U^{[\epsilon]}(z)\bigr) = U^{[\alpha]}\bigl(z + \epsilon L(z) + O(\epsilon^2)\bigr)$. Also $U^{[\alpha+\epsilon]}(z) = U^{[\epsilon]}\bigl(U^{[\alpha]}(z)\bigr) = U^{[\alpha]}(z) + \epsilon L\bigl(U^{[\alpha]}(z)\bigr) + O(\epsilon^2)$.

(d) The identity follows from the fact that U commutes with $\ln U$. It determines l_{n-2} when $n \ge 4$, because the coefficient of l_{n-2} on the left is nu_2, while the coefficient on the right is $\binom{n}{2}u_2$. Similarly, if $u_2 = \cdots = u_{k-1} = 0$ and $u_k \ne 0$, we have $l_{k-1} = u_k$ and the recurrence for $n \ge 2k$ determines l_k, l_{k+1}, $\ldots$: The left side has the form $l_{n-1} + \binom{n}{k-1}l_{n-k}u_k + \cdots$ and the right side has the form $l_{n-1} + \binom{n}{k}l_{n-k}u_k + \cdots$. In general, $l_1 = u_2$, $l_2 = u_3 - \frac{3}{2}u_2^2$, $l_3 = u_4 - 5u_2u_3 + \frac{9}{2}u_2^3$, $l_4 = u_5 - \frac{15}{2}u_2u_4 - 5u_3^2 + \frac{185}{6}u_2^2u_3 - 20u_2^4$.

(e) We have $U = \sum_m (\ln U)^m/m!$, and for fixed m the contribution to $u_n = u_{n1}$ from the mth term is $\sum l_{n_m n_{m-1}} \ldots l_{n_2 n_1} l_{n_1 n_0}$ summed over $n = n_m > \cdots > n_1 > n_0 = 1$. Now apply the result of part (b). [Cf. *Trans. Amer. Math. Soc.* **108** (1963), 457–477.]

24. (a) By (21) and exercise 20, we have $U = VDV^{-1}$ where V is the power matrix of the Schröder function and D is the diagonal matrix $(u, u^2, u^3, \ldots)$. So we may take $\ln U = V\,\mathrm{diag}(\ln u, 2\ln u, 3\ln u, \ldots)V^{-1}$. (b) The equation $WVDV^{-1} = VDV^{-1}W$ implies $(V^{-1}WV)D = D(V^{-1}WV)$. The diagonal entries of D are distinct, so $V^{-1}WV$ must be a diagonal matrix D'. Thus $W = VD'V^{-1}$, and W has the same Schröder function as U. It follows that $W_1 \ne 0$ and $W = VD^\alpha V^{-1}$, where $\alpha = (\ln W_1)/(\ln U_1)$.

25. We must have $k = l$ because $[z^{k+l-1}]\,U\bigl(V(z)\bigr) = U_{k+l-1} + V_{k+l-1} + kU_kV_l$. To complete the proof it suffices to show that $U_k = V_k$ and $U\bigl(V(z)\bigr) = V\bigl(U(z)\bigr)$ implies $U(z) = V(z)$. Suppose l is minimal with $U_l \ne V_l$, and let $n = k + l - 1$. Then we have $u_{nk} - v_{nk} = \binom{n}{l}(u_l - v_l)$; $u_{nj} = v_{nj}$ for all $j > k$; $u_{nl} = \binom{n}{k}u_k$; $u_{nj} = 0$ for $l < j < n$. Now the sum $\sum_j u_{nj}v_j = u_n + u_{nk}v_k + \cdots + u_{nl}v_l + v_n$ must be equal to $\sum_j v_{nj}u_j$; so we find $\binom{n}{l}(u_l - v_l)v_k = \binom{n}{k}v_k(u_l - v_l)$. But $\binom{k+l-1}{k} = \binom{k+l-1}{l}$ iff $k = l$.

[From this exercise and the previous one, we might suspect that $U\bigl(V(z)\bigr) = V\bigl(U(z)\bigr)$ only when one of U and V is an iterate of the other. But this is not necessarily true when U_1 and V_1 are roots of unity. For example, if $V_1 = -1$ and $U(z) = V^{[2]}(z)$, V is not an iterate of $U^{[1/2]}$, nor is $U^{[1/2]}$ an iterate of V.]

Page 658 another new answer ______ 16 May 1995

26. From $\bigl(W(qz) - W(z)\bigr)V(z) = W(z)\bigl(V(q^mz) - V(z)\bigr)$ we obtain the recurrence $W_n = \sum_{k=1}^n V_kW_{n-k}(q^{km} - q^{n-k})/(q^n - 1)$. [*J. Difference Eqs. and Applics.* **1** (1995), 57–60.]

Page 658 another new answer ______ 29 Apr 1995

27. Note first that $\delta\bigl(U(z)V(z)\bigr) = \bigl(\delta U(z)\bigr)V(z) + U(z)\bigl(\delta V(z)\bigr)$, because $t(mn) = t(m) + t(n)$. Therefore $\delta\bigl(V(z)^n\bigr) = nV(z)^{n-1}\delta V(z)$ for all $n \ge 0$, by induction on n; and this is the identity we need to show that $\delta e^{V(z)} = \sum_{n\ge0}\delta\bigl(V(z)^n/n!\bigr) = e^{V(z)}\delta V(z)$. Replacing $V(z)$ by $\ln V(z)$ in this equation gives $V(z)\,\delta\ln V(z) = \delta V(z)$; hence $\delta\bigl(V(z)^\alpha\bigr) = \delta e^{\alpha\ln V(z)} = e^{\alpha\ln V(z)}\delta\bigl(\alpha\ln V(z)\bigr) = \alpha V(z)^{\alpha-1}$ for all complex numbers α.

It follows that the desired recurrences are

(a) $W_1 = 1$, $W_n = \sum_{d\backslash n,\,d>1}\bigl((\alpha+1)t(d)/t(n) - 1\bigr)V_dW_{n/d}$;
(b) $W_1 = 1$, $W_n = \sum_{d\backslash n,\,d>1}\bigl(t(d)/t(n)\bigr)V_dW_{n/d}$;
(c) $W_1 = 0$, $W_n = V_n + \sum_{d\backslash n,\,d>1}\bigl(t(d)/t(n) - 1\bigr)V_dW_{n/d}$.

[Cf. H. W. Gould, *AMM* **81** (1974), 3–14. These formulas hold when t is any function such that $t(m)+t(n)=t(mn)$ and $t(n)=0$ iff $n=1$, but the suggested t is simplest. The method discussed here works also for power series in arbitrarily many variables; then t is the total degree of a term.]

▸**Page 659** entry for $\sqrt{5}$ ——— 26 Apr 1985

17623 $\leadsto$ 27623

Page 661 replacement for lines 1–8 ——— 01 Nov 1995

Several of the 40-digit values in Tables 1 and 2 were computed on a desk calculator by John W. Wrench, Jr., for the first edition of this book. When computer software for such calculations became available during the 1970s, all of his contributions proved to be correct. The 40-digit values of other fundamental constants can be found in Eq. 4.5.3–(39) and the answer to exercise 4.5.4–25.

Page 662 lines 7–10 ——— 01 Nov 1995

$$\begin{aligned}
H_{1/5} &= 5-\tfrac12\pi\phi^{3/2}5^{-1/4}-\tfrac54\ln 5-\tfrac12\sqrt5\ln\phi,\\
H_{2/5} &= \tfrac52-\tfrac12\pi\phi^{-3/2}5^{-1/4}-\tfrac54\ln 5+\tfrac12\sqrt5\ln\phi,\\
H_{3/5} &= \tfrac53+\tfrac12\pi\phi^{-3/2}5^{-1/4}-\tfrac54\ln 5+\tfrac12\sqrt5\ln\phi,\\
H_{4/5} &= \tfrac54+\tfrac12\pi\phi^{3/2}5^{-1/4}-\tfrac54\ln 5-\tfrac12\sqrt5\ln\phi,
\end{aligned}$$

Page 663 and the next several pages ——— 14 May 1976

see the corrections to pages 607–611 of Volume 1

Page 664 following the two lines for $f^{(n)}(x)$ ——— 03 May 1995

$f^{[n]}(x)$	nth iterate: $\bigl(n=0 \Longrightarrow x;\, f(f^{[n-1]}(x))\bigr)$	4.7
$f^{\{n\}}(x)$	nth induced function: $f^{\{n\}}(x)=f\bigl(xf^{\{n\}}(x)^n\bigr)$	4.7

Page 665 line −14 ——— 25 Mar 1995

$\|S\| \leadsto |S|$
cardinal: $\leadsto$ cardinality:

Page 665, line −7 . . . . 25 Mar 1995

$[y,z) \leadsto [y\,..\,z)$

▸**Page 666** line −4 ——— 03 Mar 1981

$\sum \leadsto 4\sum$

▸**Page 667** definition of lcm ——— 31 Jul 1997

$j=k=0 \leadsto jk=0$

Page 667 replacement for the entry that defines Pr ——— 03 Aug 1995

$\Pr\bigl(S(X)\bigr)$	probability that statement $S(X)$ is true, for random values of X	3.5, 4.2.4
$\mathrm{E}\,X$	expected value of X: $\sum_x x\Pr(X=x)$	1.2.10

Page 667 bottom of page ——— 03 Aug 1995

$\approx$	approximate equality	3.5, 4.2.2

Page 668 and following ______ 03 Jun 1995

Miscellaneous changes to the existing index of Volume 2 are collected here, including corrections and amendments to the old entries as well as new entries that are occasioned by the new material. Thus, the lines of the full index that have changed serve also as an index to the present document. However, when a correction or amendment has caused an old index entry to be deleted, the deletion is usually not indicated.

Page 688 bottom three lines ——— 01 Jun 1995

For additional definitions . . . 1966).

▸**Page 693** (the endpapers) ——— 08 Jan 1982

In some early printings, character number 54 was shown as '**;**' instead of '**:**'.