☆ THE ART OF COMPUTER PROGRAMMING ☆

☆☆☆☆☆☆☆☆ ERRATA, VOLUME THREE ☆☆☆☆☆☆☆☆

This document is a transcript of the notes I've made in my personal copy of *The Art of Computer Programming*, Volume 3, during the past 22 years. They apply to all copies of the first edition printed since March 1975.

Three levels of updates — "errors," "amendments," and "plans" — appear, 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.

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 edition of Volume 3. I didn't do that in the second edition 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 that volume 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 3 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, February 1998

Things have changed in the past two decades.

— BILL GATES (1995)

In addition to the errors listed here,
about half of the occurrences of 'which' in volumes one and three
should be changed to 'that'.

— DONALD E. KNUTH (*The Art of Computer Programming Errata et Addenda*, 1981)

☆☆☆☆☆☆☆ SORTING AND SEARCHING ☆☆☆☆☆☆☆

Page viii line 16 ——— 28 Feb 1997

book. ⇝ 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 ix new rating for exercise 3 ——— 25 Mar 1995

M50 ⇝ *HM45*

Page ix new quote for bottom of page ——— 27 May 1978

Two hours' daily exercise . . . will be enough
to keep a hack fit for his work.
— M. H. MAHON (*The Handy Horse Book*, 1865)

Page x and page xi . . . . 09 May 1995

In the next edition I plan to have new sections '6.2.5. Randomized Data Structures' and '6.6. Persistent Data Structures'.

Page 5 line 21 ——— 11 Feb 1995

5.3.1. ⇝ 5.3.1, if the records are in random order and if sorting is done by pairwise comparisons of keys.

Page 6 bottom line ——— 06 Jun 1995

face. ⇝ have had to solve in the old days when computers didn't have much random-access memory.

▸**Page 8** line 3 ——— 04 Jan 1997

roman ⇝ Roman

▸**Page 8** data ——— 02 Mar 1977

left column, line 20: mädeln ⇝ Mädeln
right column, line 26: Weiner ⇝ Wiener

Page 10 line 2 ——— 10 Aug 1995

`REACT`, `TRACE` ⇝ `REACT`, `RECTA`, `TRACE`

Page 15 replacement for the line before (8) ——— 26 May 1995

simple form noticed by O. Rodriguez [*J. de Math.* **4** (1839), 236–240]:

Page 17 line 3 before (14) ——— 09 May 1995

line −17: mysterious, and no simple direct proof is evident. ⇝ mysterious.
line −16, also (14) and (15): $J(a_1\,a_2\ldots a_n)$ ⇝ $\mathrm{ind}(a_1\,a_2\ldots a_n)$

Page 18 replacement for lines 4–29, up to the exercises ________ 14 May 1995

The desired correspondence is defined by a simple sorting procedure: Any n-tuple $(q_1, q_2, \ldots, q_n)$ can be rearranged into nonincreasing order $q_{a_1} \geq q_{a_2} \geq \cdots \geq q_{a_n}$ in a stable manner, where $a_1\,a_2\ldots a_n$ is a permutation such that $q_{a_j} = q_{a_{j+1}}$ implies $a_j < a_{j+1}$. We set $(p_1, p_2, \ldots, p_n) = (q_{a_1}, q_{a_2}, \ldots, q_{a_n})$ and then, for $1 \leq j < n$, subtract 1 from each of $p_1, \ldots, p_j$ for each j such that $a_j > a_{j+1}$. We still have $p_1 \geq p_2 \geq \cdots \geq p_n$, because p_j was strictly greater than p_{j+1} whenever $a_j > a_{j+1}$. The resulting pair $\bigl((a_1, a_2, \ldots, a_n), (p_1, p_2, \ldots, p_n)\bigr)$ satisfies (15), because the total reduction of the p's is $\operatorname{ind}(a_1\,a_2\ldots a_n)$. For example, if $n = 9$ and $(q_1, \ldots, q_9) = (3, 1, 4, 1, 5, 9, 2, 6, 5)$, we find $a_1 \ldots a_9 = 6\,8\,5\,9\,3\,1\,7\,2\,4$ and $(p_1, \ldots, p_9) = (5, 2, 2, 2, 2, 2, 1, 1, 1)$.

Conversely, we can easily go back to $(q_1, q_2, \ldots, q_n)$ when $a_1\,a_2\ldots a_n$ and $(p_1, p_2, \ldots, p_n)$ are given. (See exercise 17.) So the desired correspondence has been established, and MacMahon's index theorem has been proved.

A surprising extension of MacMahon's theorem was discovered in the 1970s by D. Foata and M. P. Schützenberger: *The number of permutations of n elements that have k inversions and index l is the same as the number that have l inversions and index k.* In fact, Foata and Schützenberger found a simple one-to-one correspondence between permutations of the first kind and permutations of the second (see exercise 25).

Page 19 line 2 of exercise 9 ________ 13 Apr 1981

its own inverse ⇝ an involution (i.e., its own inverse)

Page 21 replacement for exercise 19 ________ 14 May 1976

19. [*M28*] (C. Meyer.) When m is relatively prime to n, we know that the sequence $(m \bmod n)(2m \bmod n)\ldots\bigl((n-1)m \bmod m\bigr)$ is a permutation of $\{1, 2, \ldots, n-1\}$. Show that the number of inversions of this permutation can be expressed in terms of Dedekind sums (cf. Section 3.3.3).

▸**Page 21** first line of exercise 20 ________ 28 Apr 1996

Ellipticorum ⇝ *Ellipticarum*

Page 22 new exercise ________ 14 May 1976

▸**21.** [*M25*] (G. D. Knott.) Show that the permutation $a_1 \ldots a_n$ is obtainable with a stack, in the sense of exercise 2.2.1–5 or 2.3.1–6, if and only if $C_j \leq C_{j+1} + 1$ for $1 \leq j < n$ in the notation of exercise 7.

Page 22 new exercise ________ 17 Aug 1995

22. [*M26*] Given a permutation $a_1\,a_2\ldots a_n$ of $\{1, 2, \ldots, n\}$, let h_j be the number of indices $i < j$ such that $a_i \in \{a_j+1, a_j+2, \ldots, a_{j+1}\}$. (If $a_{j+1} < a_j$, the elements of this set "wrap around" from n to 1. When $j = n$ we use the set $\{a_n+1, a_n+2, \ldots, n\}$.) For example, the permutation $5\,9\,1\,8\,2\,6\,4\,7\,3$ leads to $h_1 \ldots h_9 = 0\,0\,1\,2\,1\,4\,2\,4\,6$. (a) Prove that $a_1\,a_2\ldots a_n$ can be reconstructed from the numbers $h_1\,h_2\ldots h_n$. (b) Prove that $h_1 + h_2 + \cdots + h_n$ is the index of $a_1\,a_2\ldots a_n$.

Page 22 new exercise ________ 27 Apr 1996

▸**23.** [*M26*] (*Russian roulette.*) A group of n condemned men who prefer probability theory to number theory might choose to commit suicide by sitting in a circle and modifying Josephus's method (exercise 2) as follows: The first prisoner holds a gun and aims it at his head; with probability p he dies and leaves the circle. Then the second man takes the gun and proceeds in the same way. Play continues cyclically, with constant probability $p > 0$, until everyone is dead.

Let $a_j = k$ if man k is the jth to die. Prove that the death order $a_1\,a_2\ldots a_n$ occurs with a probability that is a function only of n, p, and the index of the dual permutation $(n+1-a_n)\ldots(n+1-a_2)\,(n+1-a_1)$. What death order is least likely?

Page 22 new exercises ———— 13 May 1995

24. [*26*] Given integers $t(1)\,t(2)\ldots t(n)$ with $t(j) \ge j$, the *generalized index* of a permutation $a_1\,a_2\ldots a_n$ is the sum of all subscripts j such that $a_j > t(a_{j+1})$, plus the total number of inversions such that $i < j$ and $t(a_j) \ge a_i > a_j$. Thus when $t(j) = j$ for all j, the generalized index is the same as the index; but when $t(j) \ge n$ for all j it is the number of inversions. Prove that the number of permutations whose generalized index equals k is the same as the number of permutations having k inversions. [*Hint:* Show that, if we take any permutation $a_1\ldots a_{n-1}$ of $\{1,\ldots,n-1\}$ and insert the number n in all possible places, we increase the generalized index by the numbers $\{0,1,\ldots,n-1\}$ in some order.]

▸**25.** [*M30*] (Foata and Schützenberger.) (a) If $\alpha = a_1\ldots a_n$ is a permutation, let $\operatorname{ind}(\alpha)$ be its index, and let $\operatorname{inv}(\alpha)$ count its inversions. Define a one-to-one correspondence that takes each permutation α of $\{1,\ldots,n\}$ to a permutation $f(\alpha)$ that has the following two properties: (i) $\operatorname{ind}\bigl(f(\alpha)\bigr) = \operatorname{inv}(\alpha)$; (ii) for $1 \le j < n$, the number j appears to the left of $j+1$ in $f(\alpha)$ if and only if j appears to the left of $j+1$ in α. What permutation does your construction assign to $f(\alpha)$ when $\alpha = 1\,9\,8\,2\,6\,3\,7\,4\,5$? For what permutation α is $f(\alpha) = 1\,9\,8\,2\,6\,3\,7\,4\,5$? [*Hint:* If $n > 1$, write $\alpha = x_1\alpha_1x_2\alpha_2\ldots x_k\alpha_ka_n$, where $x_1, \ldots, x_k$ are all the elements $< a_n$ if $a_1 < a_n$, otherwise $x_1, \ldots, x_k$ are all the elements $> a_n$; the other elements appear in (possibly empty) strings $\alpha_1, \ldots, \alpha_k$. Compare the number of inversions of $h(\alpha) = \alpha_1x_1\alpha_2x_2\ldots\alpha_kx_k$ to $\operatorname{inv}(\alpha)$.]

(b) Use f to define another one-to-one correspondence g having the following two properties: (i) $\operatorname{ind}\bigl(g(\alpha)\bigr) = \operatorname{inv}(\alpha)$; (ii) $\operatorname{inv}\bigl(g(\alpha)\bigr) = \operatorname{ind}(\alpha)$. [*Hint:* Consider inverse permutations.]

Page 22 new exercise ———— 06 Jun 1995

26. [*M25*] What is the statistical correlation coefficient between the number of inversions and the index of a random permutation? (Cf. Eq. 3.3.2–(24).)

Page 22 new exercises ———— 28 Apr 1996

27. [*25*] (R. W. Floyd, 1983.) If $a_1\,a_2\,\ldots\,a_n$ is a permutation of $\{1,2,\ldots,n\}$, its *total displacement* is defined to be $\sum_{j=1}^{n}|a_j - j|$. Find upper and lower bounds for total displacement in terms of the number of inversions.

28. [*28*] If $\pi = a_1\,a_2\,\ldots\,a_n$ and $\pi' = a_1'\,a_2'\,\ldots\,a_n'$ are permutations of $\{1,2,\ldots,n\}$, their product $\pi\pi'$ is $a'_{a_1}\,a'_{a_2}\,\ldots\,a'_{a_n}$. Let $\operatorname{inv}(\pi)$ denote the number of inversions, as in exercise 25. Show that $\operatorname{inv}(\pi\pi') \le \operatorname{inv}(\pi) + \operatorname{inv}(\pi')$, and that equality holds if and only if $\pi\pi'$ is "below" π' in the sense of exercise 12.

Page 22 add a footnote to the line after (2) ———— 19 Apr 1977

A permutation $\rightsquigarrow$ A permutation*

* Sometimes called a "permatution."

Page 23 top half ——— 15 May 1995

lines 6–7: was known ... (c. 300 A.D.) $\rightsquigarrow$ has been known for a long time. The Hebrew Book of Formation (c. 400 A.D.)
line 9: obtain numbers which $\rightsquigarrow$ compute what
lines 10–11: *Sefer* ... cf. also $\rightsquigarrow$ *Sefer Yetzirah*, end of Chapter 4. Cf.
lines 13–15: 494–496. The Book ... creation.] $\rightsquigarrow$ 494–496; Aryeh Kaplan, *Sefer Yetzirah* (York Beach, Maine: Samuel Weiser, 1993).]

Page 23 lines 17 and 22 ——— 18 Oct 1979

Anuyogadvarā $\rightsquigarrow$ *Anuyogadvāra*

▸**Page 23** line −9 ——— 05 Jun 1975

45885 $\rightsquigarrow$ 45855

Page 24 lines 7–8 ——— 12 Jul 1996

Treatise of Algebra **2** (Oxford, 1685), 117–118 $\rightsquigarrow$ *Discourse of Combinations* (Oxford, 1685), Chapter 2 (published with his *Treatise of Algebra*)

▸**Page 24** line 13 ——— 28 Feb 1978

(1965 $\rightsquigarrow$ (1965)

Page 27 lines −10 and −9 ——— 29 Apr 1996

proved by Foata in 1965. $\rightsquigarrow$ proved constructively by Foata [*Comptes Rendus Acad. Sci. Paris* **258** (1964), 1672–1675].

▸**Page 32** line 3 of exercise 15 ——— 02 Oct 1997

$n_1 + n_2 + \cdots + n_m = m \rightsquigarrow n_1 + n_2 + \cdots + n_m = n$

▸**Page 34** line 12, in the determinant ——— 27 May 1978

$1 - a_{mn}x_m \rightsquigarrow 1 - a_{mm}x_m$

Page 34 new exercises ——— 17 May 1995

21. [*M21*] Given n_1, ..., n_m, and $d \ge 0$, how many permutations $a_1\,a_2 \ldots a_n$ of the multiset $\{n_1\cdot 1, \ldots, n_m\cdot m\}$ satisfy $a_{j+1} \ge a_j - d$ for $1 \le j < n = n_1 + \cdots + n_m$?

22. [*HM30*] Let $P(x_1^{n_1}\ldots x_m^{n_m})$ denote the set of all permutations of the multiset $\{n_1\cdot x_1, \ldots, n_m\cdot x_m\}$, and let $P_0(x_0^{n_0}x_1^{n_1}\ldots x_m^{n_m})$ be the subset of $P(x_0^{n_0}x_1^{n_1}\ldots x_m^{n_m})$ in which the first n_0 elements are $\ne x_0$.

(a) Given a number t with $1 \le t < m$, find a one-to-one correspondence between $P(1^{n_1}\ldots m^{n_m})$ and the set of all ordered pairs of permutations that belong respectively to $P_0(0^k 1^{n_1}\ldots t^{n_t})$ and $P_0(0^k(t+1)^{n_{t+1}}\ldots m^{n_m})$, for some $k \ge 0$. [*Hint:* For each $\pi = a_1\ldots a_n \in P(1^{n_1}\ldots m^{n_m})$, let $l(\pi)$ be the permutation obtained by replacing $t+1$, ..., m by 0 and erasing all 0s in the last $n_{t+1} + \cdots + n_m$ positions; similarly, let $r(\pi)$ be the permutation obtained by replacing 1, ..., t by 0 and erasing all 0s in the first $n_1 + \cdots + n_t$ positions.]

(b) Prove that the number of permutations of $P_0(0^{n_0}1^{n_1}\ldots m^{n_m})$ whose two-line form has p_j columns $\genfrac{}{}{0pt}{}{0}{j}$ and q_j columns $\genfrac{}{}{0pt}{}{j}{0}$ is

$$\frac{\bigl|P(x_1^{p_1}\ldots x_m^{p_m}y_1^{n_1-p_1}\ldots y_m^{n_m-p_m})\bigr|\bigl|P(x_1^{q_1}\ldots x_m^{q_m}y_1^{n_1-q_1}\ldots y_m^{n_m-q_m})\bigr|}{\bigl|P_0(0^{n_0}1^{n_1}\ldots m^{n_m})\bigr|}.$$

(c) Let $w_1, \ldots, w_m, z_1, \ldots, z_m$ be complex numbers on the unit circle. Define the weight $w(\pi)$ of a permutation $\pi \in P(1^{n_1} \ldots m^{n_m})$ as the product of the weights of its columns in two-line form, where the weight of $\substack{j\\k}$ is w_j/w_k if j and k are both $\le t$ or both $> t$, otherwise it is z_j/z_k. Prove that the sum of $w(\pi)$ over all $\pi \in P(1^{n_1} \ldots m^{n_m})$ is

$$\sum_{k\ge 0} \frac{k!^2 (n_{\le t} - k)!\,(n_{>t} - k)!}{n_1! \ldots n_m!} \left| \sum \binom{n_1}{p_1} \cdots \binom{n_m}{p_m} \left(\frac{w_1}{z_1}\right)^{p_1} \cdots \left(\frac{w_m}{z_m}\right)^{p_m} \right|^2,$$

where $n_{\le t}$ is $n_1 + \cdots + n_t$, $n_{>t}$ is $n_{t+1} + \cdots + n_m$, and the inner sum is over all $(p_1, \ldots, p_m)$ such that $p_{\le t} = p_{>t} = k$.

23. [*M23*] A strand of DNA can be thought of as a word on a four-letter alphabet. Suppose we copy a strand of DNA and break it completely into one-letter bases, then recombine those bases at random. If the resulting strand is placed next to the original, prove that the number of places in which they differ is more likely to be even than odd. [*Hint:* Apply the previous exercise.]

Page 34 new exercise ________ 17 Aug 1995

24. [*25*] Consider any relation R that might hold between two unordered pairs of letters; if $\{w,x\}R\{y,z\}$ we say $\{w,x\}$ *preserves* $\{y,z\}$, otherwise $\{w,x\}$ *moves* $\{y,z\}$. The operation of *transposing* $\substack{w\ x\\y\ z}$ with respect to R replaces $\substack{w\ x\\y\ z}$ by $\substack{x\ w\\y\ z}$ or $\substack{x\ w\\z\ y}$, according as the pair $\{w,x\}$ preserves or moves the pair $\{y,z\}$, assuming that $w \ne x$ and $y \ne z$; if $w = x$ or $y = z$ the transposition always produces $\substack{x\ w\\z\ y}$.

(a) The operation of *sorting* a two-line array $\binom{x_1\,\ldots\,x_n}{y_1\,\ldots\,y_n}$ with respect to R repeatedly finds the largest x_j such that $x_j > x_{j+1}$ and transposes columns j and $j+1$, until eventually $x_1 \le \cdots \le x_n$. (We do not require $x_1 \ldots x_n$ to be a permutation of $y_1 \ldots y_n$.) Prove that for every $x \in \{x_1, \ldots, x_n\}$ there is a unique $y \in \{y_1, \ldots, y_n\}$ such that $\mathrm{sort}\binom{x_1\,\ldots\,x_n}{y_1\,\ldots\,y_n} = \mathrm{sort}\binom{x\,x_2'\,\ldots\,x_n'}{y\,y_2'\,\ldots\,y_n'}$ for some $x_2', y_2', \ldots, x_n', y_n'$.

(b) Let $\binom{w_1\,\ldots\,w_k}{y_1\,\ldots\,y_k} \circledR \binom{x_1\,\ldots\,x_l}{z_1\,\ldots\,z_l}$ denote the result of sorting $\binom{w_1\,\ldots\,w_k\ \ x_1\,\ldots\,x_l}{y_1\,\ldots\,y_k\ \ z_1\,\ldots\,z_l}$ with respect to R. For example, if R is always true, $\circledR$ is simply juxtaposition; if R is always false, $\circledR$ is the intercalation product. Generalize Theorem A by proving that every permutation π of a multiset M has a unique representation of the form

$$\pi = (x_{11} \ldots x_{1n_1}\, y_1) \circledR \bigl((x_{21} \ldots x_{2n_2}\, y_2) \circledR \cdots \circledR (x_{t1} \ldots x_{tn_t}\, y_t)\bigr)$$

satisfying (16), if we define cycle notation by letting (11) stand for $(x_2 \ldots x_n\, x_1)$ instead of (10). For example, suppose $\{w,x\}R\{y,z\}$ means that w, x, y, and z are distinct; then it turns out that the factorization of (12) analogous to (17) is

$$(d\,d\,b\,c\,a) \circledR \bigl((c\,b\,b\,a) \circledR \bigl((c\,d\,b) \circledR \bigl((d\,b) \circledR (d)\bigr)\bigr)\bigr).$$

(The operation $\circledR$ does not always obey the associative law; parentheses in the generalized factorization should be nested from right to left.)

Page 35 begins an important notational change ________ 13 May 1995

line 2: k ascending ⤳ k "descents" $a_j > a_{j+1}$, thus exactly $k+1$ ascending
line 6: 5.1.4–22. ⤳ 5.1.4–22. The angle brackets in $\left\langle{n \atop k}\right\rangle$ remind us of the ">" sign in the definition of a descent. Of course $\left\langle{n \atop k}\right\rangle$ is also the number of permutations that have k "ascents" $a_j < a_{j+1}$.

lines 9–10: k runs, ... number of runs $\rightsquigarrow$ k descents, exactly $k+1$ of these new permutations will have k descents; the remaining $n-1-k$ will have $k+1$, since we increase the number of descents

replacement for lines 15–20:

all but the second and last of these have two descents instead of one. Therefore we have the recurrence relation

$$\left\langle {n \atop k} \right\rangle = (k+1)\left\langle {n-1 \atop k} \right\rangle + (n-k)\left\langle {n-1 \atop k-1} \right\rangle, \quad \text{integer } n > 0, \text{ integer } k. \tag{2}$$

By convention we set

$$\left\langle {0 \atop k} \right\rangle = \delta_{k0}\,, \tag{3}$$

saying that the null permutations has no descents. The reader may find it interesting

replacement for (5) to the bottom of the page:

$$\left\langle {n \atop 0} \right\rangle = 1\,, \qquad \left\langle {n \atop n} \right\rangle = 0\,; \tag{5}$$

$$\left\langle {n \atop n-1} \right\rangle = 1\,, \qquad \text{for } n \geq 1. \tag{6}$$

Eq. (6) follows from (5) because of a general rule of symmetry,

$$\left\langle {n \atop k} \right\rangle = \left\langle {n \atop n-1-k} \right\rangle, \qquad \text{for } n \geq 1, \tag{7}$$

which comes from the fact that each nonnull permutation $a_1\, a_2\, \ldots\, a_n$ having k descents has $n-1-k$ ascents.

Page 36 replacement for (8) and the following clause ——— 13 May 1995

$$\sum_k \left\langle {n \atop k} \right\rangle \binom{m+k}{n} = m^n\,, \qquad n \geq 0, \tag{8}$$

which was discovered by the Chinese mathematician Li Shan-Lan and published in 1867. [See J.-C. Martzloff, *Histoire des Mathématiques Chinoises* (Paris, 1988), 320-325.] Li Shan-Lan's identity follows from the properties of sorting:

Page 36 line 3 before (10) ——— 13 May 1995

$n+1-k$. $\rightsquigarrow$ $n-k$ and use (7).

Page 36 replacement for (11) ——— 13 May 1995

$$x^n = \left\langle {n \atop 0} \right\rangle \binom{x}{n} + \left\langle {n \atop 1} \right\rangle \binom{x+1}{n} + \cdots + \left\langle {n \atop n-1} \right\rangle \binom{x+n-1}{n}, \quad n \geq 1. \tag{11}$$

Page 36 replacement for bottom two lines and top three of page 37 13 May 1995

Setting $x = 1$ in (11) proves again that $\left\langle {n \atop n-1} \right\rangle = 1$, since the binomial coefficients vanish in all but the last term. Setting $x = 2$ yields

$$\left\langle {n \atop n-2} \right\rangle = \left\langle {n \atop 1} \right\rangle = 2^n - n - 1\,, \qquad n \ge 1. \tag{12}$$

Page 37 replacement for (13) 13 May 1995

$$\begin{aligned}\left\langle {n \atop k} \right\rangle &= (k+1)^n - k^n\binom{n+1}{1} + (k-1)^n\binom{n+1}{2} - \cdots + (-1)^k 1^n \binom{n+1}{k}\\ &= \sum_{j=0}^{k}(-1)^j\binom{n+1}{j}(k+1-j)^n, \qquad n \ge 0,\ k \ge 0.\end{aligned} \tag{13}$$

Page 37 replacement for (14) 13 May 1995

$$g_n(z) = \sum_k \left\langle {n \atop k-1} \right\rangle z^k/n!\,, \tag{14}$$

Page 37 Table 1 13 May 1995

The numbers in columns headed $\left\langle {n \atop 1} \right\rangle$ through $\left\langle {n \atop 7} \right\rangle$ should be moved left, so that they're now headed $\left\langle {n \atop 0} \right\rangle$ through $\left\langle {n \atop 6} \right\rangle$. The column of zeros previously headed $\left\langle {n \atop 0} \right\rangle$ should move to the rightmost column, headed $\left\langle {n \atop 7} \right\rangle$.

Page 38 middle of page 13 May 1995

change the period to a comma in (18), then replace the next four lines by:
when $n \ge 1$. The super generating function

$$g(z,x) = \sum_{n \ge 0} \frac{g_n(z)x^n}{z} = \sum_{k,n \ge 0} \left\langle {n \atop k} \right\rangle z^k x^n/n! \tag{19}$$

is therefore equal to

$$\sum_{k,n\ge 0} \frac{\bigl((z-1)x\bigr)^n}{(z-1)^k} \left\{ {n \atop k} \right\} \frac{k!}{n!} = \sum_{k \ge 0}\left(\frac{e^{(z-1)x}-1}{z-1}\right)^k = \frac{(1-z)}{e^{(z-1)x}-z}\,; \tag{20}$$

Page 39 replacement for (23) 13 May 1995

$$q_{km} = \left(\left\langle {m \atop 0} \right\rangle + \cdots + \left\langle {m \atop k-1} \right\rangle\right) \Big/ m!\,. \tag{23}$$

Page 40 replacement for (24) and the next two lines 13 May 1995

$$L_k = \frac{1}{1!}\left\langle {1 \atop k-1} \right\rangle + \frac{1}{2!}\left\langle {2 \atop k-1} \right\rangle + \frac{1}{3!}\left\langle {3 \atop k-1} \right\rangle + \cdots = \sum_{m \ge 1}\left\langle {m \atop k-1} \right\rangle \frac{1}{m!}\,. \tag{24}$$

Since $\left\langle {0 \atop k-1} \right\rangle = 0$ except when $k = 1$, L_k turns out to be the coefficient of z^{k-1} in the generating function $g(z,1) - 1$ (see Eq. (19)), so we have

Page 45 in the left sides of the displayed formulas of exercise 2 —— 13 May 1995

$\binom{k-1}{n-q} \rightsquigarrow \binom{k}{n-q}$ $\qquad$ $\binom{k}{m} \rightsquigarrow \binom{k+1}{m}$

Page 45 in the right side of the displayed formula of exercise 6 —— 13 May 1995

$j^n \rightsquigarrow (j+1)^n$

▸**Page 46** line 1 of exercise 11 —— 03 Oct 1997

$p_{km} \rightsquigarrow P_{km}$ (twice)

Page 46 replacement for exercise 18 —— 17 May 1995

18. [*M28*] Among all $n!$ sequences $b_1 b_2 \ldots b_n$ such that each b_j is an integer in the range $0 \le b_j \le n-j$, how many have (a) exactly k descents (i.e., k occurrences of $b_j > b_{j+1}$)? (b) exactly k distinct elements?

Page 48 line 4 —— 13 May 1995

$\lceil U_1 + \cdots + U_n \rceil \rightsquigarrow \lfloor U_1 + \cdots + U_n \rfloor$

Page 48 new exercise —— 07 Sep 1995

26. [*M20*] Let ϑ be the operation $z\frac{d}{dz}$, which multiplies the coefficient of z^n in a generating function by n. Show that the result of applying ϑ to $1/(1-z)$ repeatedly, m times, can be expressed in terms of Eulerian numbers.

Page 48 new exercise —— 28 Apr 1997

▸**27.** [*M21*] An *increasing forest* is a forest in which the nodes are labeled $\{1, 2, \ldots, n\}$ in such a way that parents have smaller numbers than their children. Show that $\left\langle {n \atop k} \right\rangle$ is the number of n-node increasing forests with $k+1$ leaves.

Page 48 lines 3 and 4 after (1) —— 22 May 1995

D. E. Rutherford ... 1968) $\rightsquigarrow$ Bruce E. Sagan, *The Symmetric Group* (Pacific Grove, Calif.: Wadsworth & Brooks/Cole, 1991)

Page 50 (6) and (7) —— 21 May 1995

in (6): $a = b = \infty \rightsquigarrow a = b = 0$ or $a = b = \infty$
in line 1 of (7): if $\rightsquigarrow$ if and only if

▸**Page 58** line 4 —— 28 Feb 1978

$\min\,\{a_1\ a_2, \ldots, a_n\} \rightsquigarrow \min\{a_1, a_2, \ldots, a_n\}$

Page 63 lines 7 and 8 —— 21 May 1995

so we have ... Thrall). *The number* $\rightsquigarrow$ so we have derived the following result due to J. S. Frame, G. de B. Robinson, and R. M. Thrall [*Canadian J. Math.* **6** (1954), 316–318]:

Theorem H. *The number*

Page 63 lines 17–19 —— 21 May 1995

All known proofs ... hooks). $\rightsquigarrow$ See exercises 35, 36, and 38 for the simplest known proofs of Theorem H that correctly use combinatorial properties of hooks.

Page 63 line 7 before (37) ——— 21 May 1995

S's and X's $\rightsquigarrow$ n S's and n X's

▸**Page 63** line −4 ——— 27 May 1978

S's never exceeds the number of X's $\rightsquigarrow$ X's never exceeds the number of S's

Page 64 lines 5–8 after (38) ——— 14 May 1976

Curiously ... situation to the $\rightsquigarrow$ An interesting one-to-one correspondence between such permutations and binary trees, more direct than the roundabout method via Algorithm I that we have used here, has been found by D. Rotem [*Inf. Proc. Letters* **4** (1975), 58–61]; similarly there is a

Page 65 line 1 after (40) ——— 22 May 1995

small n. $\rightsquigarrow$ small n. The values for $n \geq 0$ are 1, 1, 2, 4, 10, 26, 76, 232, 764, 2620, 9496, ...

▸**Page 65** Eq. (41) ——— 19 Dec 1996

$$\sum_{k\geq 0} \rightsquigarrow \sum_{k=0}^{\lfloor n/2\rfloor}$$

Page 65 line −9 ——— 21 May 1995

this sum cannot ... knows, $\rightsquigarrow$ this sum has no simple closed form (unless we choose to regard the Hermite polynomial $i^n 2^{-n/2} H_n(-i/\sqrt{2}\,)$ as simple),

Page 67 following (50) ——— 27 May 1978

lines 2–4: we find ... ∞. Euler's $\rightsquigarrow$ Euler's
line 5: in this case, since $\rightsquigarrow$ since
lines 7–8 (the two lines following (51)): n; this ... we have proved that $\rightsquigarrow$ n. The derivative $g^{(m)}(y)$ is a polynomial in y times e^{-2y^2}, hence $R_m = O(n^{(t+1-m)/4}) \int_{-\infty}^{+\infty} |g^{(m)}(y)|\,dy = O(n^{(t+1-m)/4})$. Furthermore if we replace α and β by $-\infty$ and $+\infty$ in the right-hand side of (50), we make an error of at most $O\bigl(\exp(-2n^{\epsilon})\bigr)$ in each term. Thus

Page 67 on the line following (53) ——— 14 May 1976

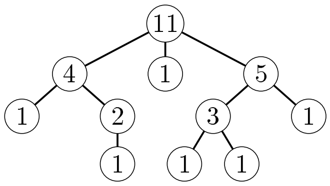
In principle, this $\rightsquigarrow$ Actually the O terms here should have an extra 9ϵ in the exponent, but our manipulations make it clear that this 9ϵ would disappear if we had carried further accuracy in the intermediate calculations. In principle, the method we have used

Page 67 lines 3 and 4 after (53) ——— 30 Apr 1996

For further examples and extensions of the technique used to derive (53), $\rightsquigarrow$ (new paragraph) The method we have used to derive (53) is an extremely useful technique for asymptotic analysis that was introduced by P. S. Laplace [*Mémoires Acad. Sci. Paris* (1782), 1–88]; it is discussed under the name "trading tails" in *CMath*, §9.4. For further examples and extensions of tail-trading,

Page 70 exercise 20 ______ 18 May 1995

line 1: binary tree $\leadsto$ tree
replacement for bottom two lines:



is $11!\,/\,11 \cdot 4 \cdot 1 \cdot 5 \cdot 1 \cdot 2 \cdot 3 \cdot 1 \cdot 1 \cdot 1 \cdot 1 = 10 \cdot 9 \cdot 8 \cdot 7 \cdot 6$.

Page 72 replacement for exercise 28 ______ 15 Nov 1978

28. [*M43*] Prove that the average length of the longest increasing subsequence of a random permutation on $\{1, 2, \ldots, n\}$ is asymptotically $2\sqrt{n}$. (This is the average length of row 1 in the correspondence of Theorem A.)

Page 72 new exercises ______ 30 Apr 1996

32. [*HM21*] Prove that t_n is the expected value of X^n, when X is a normal deviate with mean 1 and variance 1.

33. [*M25*] (O. H. Mitchell, 1881.) True or false: $\Delta(n_1, n_2, \ldots, n_m)/\Delta(1, 2, \ldots, m)$ is an integer when n_1, n_2, $\ldots$, n_m are integers.

Page 72 new exercise ______ 19 May 1995

34. [*25*] (T. Nakayama, 1940.) Prove that if a tableau shape contains a hook of length ab, it contains a hook of length a.

Page 72 new exercises ______ 20 May 1995

▸**35.** [*30*] (A. P. Hillman and R. M. Grassl, 1976.) An arrangement of nonnegative integers p_{ij} in a tableau shape is called a *plane partition of m* if $\sum p_{ij} = m$ and

$$p_{i1} \ge \cdots \ge p_{ir_i}, \qquad p_{1j} \ge \cdots \ge p_{c_j j}, \qquad \text{for } 1 \le i \le c_1,\ 1 \le j \le r_1,$$

when there are r_i cells in row i and c_j cells in column j. It is called a *reverse plane partition* if instead

$$p_{i1} \le \cdots \le p_{ir_i}, \qquad p_{1j} \le \cdots \le p_{c_j j}, \qquad \text{for } 1 \le i \le c_1,\ 1 \le j \le r_1.$$

Consider the following algorithm, which operates on reverse plane partitions of a given shape and constructs another array of numbers q_{ij} having the same shape:

G1. [Initialize.] Set $q_{ij} \leftarrow 0$ for $1 \le j \le r_i$ and $1 \le i \le c_1$. Then set $j \leftarrow 1$.

G2. [Find nonzero cell.] If $p_{c_j j} > 0$, set $i \leftarrow c_j$, $k \leftarrow j$, and go on to step G3. Otherwise if $j < r_1$, increase j by 1 and repeat this step. Otherwise stop (the p array is now zero).

G3. [Decrease p.] Decrease p_{ik} by 1.

G4. [Move up or right.] If $i > 1$ and $p_{(i-1)k} > p_{ik}$, decrease i by 1 and return to G3. Otherwise if $k < r_i$, increase k by 1 and return to G3.

G5. [Increase q.] Increase q_{ij} by 1 and return to G2. ▌

Prove that this construction defines a one-to-one correspondence between reverse plane partitions of m and solutions of the equation

$$m = \sum h_{ij} q_{ij} ,$$

where the numbers h_{ij} are the hook lengths of the shape, by designing an algorithm that recomputes the p's from the q's.

36. [*HM27*] (R. P. Stanley, 1971.) (a) Prove that the number of reverse plane partitions of m in a given shape is $[z^m]\,1/\prod(1-z^{h_{ij}})$, where the numbers h_{ij} are the hook lengths of the shape. (b) Derive Theorem H from this result. [*Hint:* What is the asymptotic number of partitions as $m \to \infty$?]

37. [*M20*] (P. A. MacMahon, 1912.) What is the generating function for all plane partitions? (The coefficient of z^m should be the total number of plane partitions of m when the tableau shape is unbounded.)

Page 72 new exercise ______ 21 May 1995

▸**38.** [*M30*] (Greene, Nijenhuis, and Wilf, 1979.) We can construct a directed acyclic graph on the cells T of any given tableau shape by letting arcs run from each cell to the other cells in its hook; the outdegree of cell (i,j) will then be $d_{ij} = h_{ij} - 1$, where h_{ij} is the hook length. Suppose we generate a random path in this digraph by choosing a random starting cell (i,j) and choosing further arcs at random, until coming to a corner cell from which there is no exit. Each random choice is made uniformly.

(a) Let (a,b) be a corner cell of T, and let $I = \{i_0,\ldots,i_k\}$ and $J = \{j_0,\ldots,j_l\}$ be sets of rows and columns with $i_0 < \cdots < i_k = a$ and $j_0 < \cdots < j_l = b$. The digraph contains $\binom{k+l}{k}$ paths whose row and columns sets are respectively I and J; let $P(I,J)$ be the probability that the random path is one of these. Prove that $P(I,J) = 1/(n\,d_{i_0 b}\ldots d_{i_{k-1} b}\,d_{a j_0}\ldots d_{a j_{l-1}})$, where $n = |T|$.

(b) Let $f(T) = n!/\prod h_{ij}$. Prove that the random path ends at corner (a,b) with probability $f(T \setminus \{(a,b)\})/f(T)$.

(c) Show that the result of (b) proves Theorem H and also gives us a way to generate a random tableau of shape T, with all $f(T)$ tableaux equally likely.

Page 72 new exercise ______ 16 May 1996

39. [*HM43*] Suppose a random Young tableau has been constructed by successively placing the numbers 1, 2, ..., n in such a way that each possibility is equally likely when a new number is placed. For example, the tableau (1) would be obtained with probability $\frac{1}{1}\cdot\frac{1}{2}\cdot\frac{1}{2}\cdot\frac{1}{3}\cdot\frac{1}{3}\cdot\frac{1}{3}\cdot\frac{1}{4}\cdot\frac{1}{3}\cdot\frac{1}{3}\cdot\frac{1}{4}\cdot\frac{1}{4}\cdot\frac{1}{5}\cdot\frac{1}{4}\cdot\frac{1}{5}\cdot\frac{1}{4}$ using this procedure.

Prove that, with high probability, the resulting shape $(n_1, n_2, \ldots, n_m)$ will have $m \approx \sqrt{6n}$ and $\sqrt{k} + \sqrt{n_{k+1}} \approx \sqrt{m}$ for $0 \le k \le m$.

Page 72 new exercises ______ 07 Sep 1996

[Move the former exercise 5.2.1–39 from page 105 to this page, and call it exercise 40 instead. Then add the following three more exercises:]

▸**41.** [*30*] (*Disorder in a genome.*) The DNA of *Lobelia fervens* has genes occurring in the sequence $g_7^R g_1 g_2 g_4 g_5 g_3 g_6^R$, where g_7^R stands for the left-right reflection of g_7; the same genes occur in tobacco plants, but in the order $g_1g_2g_3g_4g_5g_6g_7$. Show that five "flip" operations on substrings are needed to get from $g_1g_2g_3g_4g_5g_6g_7$ to $g_7^R g_1 g_2 g_4 g_5 g_3 g_6^R$. (A flip takes $\alpha\beta\gamma$ to $\alpha\beta^R\gamma$, when α, β, and γ are strings.)

42. [*35*] Continuing the previous exercise, show that at most $n+1$ flips are needed to sort any rearrangement of $g_1 g_2 \ldots g_n$. Construct examples that require $n+1$ flips, for all $n > 3$.

43. [*M37*] Show that the average number of flips required to sort a random arrangement of n genes is greater than $n - H_n$, if all $2^n\, n!$ genome rearrangements are equally likely.

Page 72 more to come? . . . . 18 May 1995

There has been a tremendous explosion in knowledge about tableaux since Section 5.1.4 was first written in 1968. I won't be able to decide how much of this literature should be cited until I finish writing the group-theoretical parts of Volume 4.

▸**Page 76** line −7 ——— 18 Oct 1979

$p(n)$ ⇝ $p(N)$

▸**Page 79,** last line before exercises ——— 21 Sep 1976

Feurzig ⇝ Feurzeig

Page 84 future change . . . . 27 May 1995

I will adopt the traditional name "Shellsort" for the "diminishing increment sort" when I revise this chapter, but I'll still call it Algorithm D (because Algorithm S already exists).

Page 85 change of notation ——— 28 May 1995

lines 4 and 10: $h_t, h_{t-1}, \ldots, h_1$ ⇝ $h_{t-1}, h_{t-2}, \ldots, h_0$
line 4: h_1 equals ⇝ h_0 equals
line 11: $h_1 = 1$ ⇝ $h_0 = 1$
line 13: $s = t, t-1, \ldots, 1$ ⇝ $s = t-1, t-2, \ldots, 0$
line −8: `ENT3 T` ⇝ `ENT3 T-1`
line −8: $s \leftarrow t$ ⇝ $s \leftarrow t-1$

Page 86 change of notation, continued ——— 28 May 1995

line 13: `J3P` ⇝ `J3NN`
line 13: $t \ge s \ge 1$ ⇝ $t > s \ge 0$
line 15: $h_t, \ldots, h_1$ ⇝ $h_{t-1}, \ldots, h_0$
line 22: formulas numbered (8) ⇝ formula numbered (12)
line 26: $h_1 + \cdots + h_t$ ⇝ $h_0 + h_1 + \cdots + h_{t-1}$
line −11: h_t-sorting $\ldots h_1$-sorting ⇝ h_{t-1}-sorting, followed by h_{t-2}-sorting, $\ldots$, followed by h_0-sorting.
line −8: $h_2 = 2$ and $h_1 = 1$ ⇝ $h_1 = 2$ and $h_0 = 1$.

Page 89 change of notation, continued ——— 28 May 1995

line 14: $h_t, \ldots, h_1$ ⇝ $h_{t-1}, \ldots, h_1, h_0$
line 15: $t > s \ge 1$ ⇝ $t-1 > s \ge 0$
line 17, under the $\sum$: $t \ge s \ge 1$ ⇝ $t > s \ge 0$
line 18: $h_{t+1} = Nh_t$ ⇝ $h_t = Nh_{t-1}$

Page 90 change of notation, continued — 28 May 1995

lines -17 and -10: h_t, …, h_1 $\rightsquigarrow$ h_{t-1}, …, h_1, h_0
line -13: $h_2 \rightsquigarrow h_1$ (twice)

Page 91 change of notation, continued — 28 May 1995

line -2: $h_s = 2^s - 1 \rightsquigarrow h_s = 2^{s+1} - 1$
line -1: $1 \le s \le t \rightsquigarrow 0 \le s < t$

Page 91 lines -7 through -3 — 28 May 1995

Moreover, … cases: $\rightsquigarrow$ Therefore the average-case analysis of Algorithm D for general increments h_{t-1}, …, h_0 has baffled everyone so far when $t > 3$. Moreover, there's no obvious way to find the worst case when N and h_{t-1}, …, h_0 are given. We can, however, derive several facts about the approximate maximum running time when the increments have certain forms:

Page 92 change of notation, continued — 28 May 1995

lines 2 and 4: $B_t + \cdots + B_1 \rightsquigarrow B_{t-1} + \cdots + B_0$
line 2: $t \ge s \ge t/2 \rightsquigarrow t > s \ge t/2$

Page 92 lines 16–19 — 12 Oct 1996

algorithm. Unfortunately, … consider the *total* $\rightsquigarrow$
algorithm; see exercises 30 and 31. Unfortunately, … large. The best sequences for real-world N appear to satisfy $h_s \approx \rho^s$, where the ratio $\rho \approx h_{s+1}/h_s$ is roughly independent of s but may depend on N.

We have observed that it is unwise to choose increments in such a way that each is a divisor of all its predecessors; but we should not conclude that the best increments are relatively prime to all of their predecessors. Indeed, every element of a file that is gh-sorted and gk-sorted with $h \perp k$ has at most $\frac{1}{2}(h-1)(k-1)$ inversions when we are g-sorting. (See exercise 21.) Pratt's sequence $\{2^p 3^q\}$ wins as $N \to \infty$ by exploiting this fact, but it grows too slowly for practical use.

Janet Incerpi and Robert Sedgewick [*J. Comp. Syst. Sci.* **31** (1985), 210–224] found a way to have the best of both worlds, by showing how to construct a sequence of increments for which $h_s \approx \rho^s$ yet each increment is the gcd of two of its predecessors. Given any number $\rho > 1$, their idea is to first define a *base sequence* a_1, a_2, … by letting a_k be the least integer $\ge \rho^k$ such that $a_j \perp a_k$ for $1 \le j < k$. If $\rho = 2.5$, for example, the base sequence is

$$a_1, a_2, a_3, \ldots = 3,\ 7,\ 16,\ 41,\ 101,\ 247,\ 613,\ 1529,\ 3821,\ 9539,\ \ldots.$$

Now they define the increments

$$h_0, h_1, h_2, \ldots = 1;\ a_1;\ a_2,\ a_1a_2;\ a_1a_3,\ a_2a_3,\ a_1a_2a_3;\ \ldots$$

by setting $h_0 = 1$ and

$$h_s = h_{s-r}a_r \qquad \text{for } \binom{r}{2} < s \le \binom{r+1}{2}. \tag{8}$$

For example, when $\rho = 2.5$ the increments are

$$1,\ 3,\ 7,\ 21,\ 48,\ 112,\ 336,\ 861,\ 1968,\ 4592,\ 13776,\ 33936,\ 86961,\ 198768,\ \ldots\ .$$

The crucial point is that we can turn recurrence (8) around:

$$h_s = h_{r+s}/a_r = h_{\binom{r}{2}}/a_{\binom{r}{2}-s} \qquad \text{for } \binom{r-1}{2} \le s \le \binom{r}{2}. \tag{9}$$

Therefore, by the argument in the previous paragraph, the number of inversions per element when we are h_0-sorting, h_1-sorting, … is at most

$$f(a_2,a_1); f(a_3,a_2), f(a_3,a_1); f(a_4,a_3), f(a_4,a_2), f(a_4,a_1); \ldots \tag{10}$$

here $f(h,k) = \frac{1}{2}(h-1)(k-1)$. If $\rho^{t-1} \le N < \rho^t$, the total number B of moves is at most N times the sum of the first t elements of this sequence. Therefore (see exercise 39) we can prove that the worst case running time is much better than order $N^{1.5}$:

Theorem I. *The running time for Algorithm D is* $O(Ne^{c\sqrt{\ln n}})$ *when the increments* h_s *are defined by* (8). *Here* $c = \sqrt{8 \ln \rho}$ *and the constant implied by* O *depends on* ρ. ▌

This asymptotic upper bound as $N \to \infty$ is not especially important, because Pratt's sequence does better. The main point of Theorem I is that a sequence of increments with the practical growth rate $h_s \approx \rho^s$ can have a guaranteed running time that is $O(N^{1+\epsilon})$ for arbitrarily small $\epsilon > 0$, when any value $\rho > 1$ is given.

Let's consider practical sizes of N more carefully by looking at the *total*

Page 93 change in notation, continued ———— 28 May 1995

lines 2, 8, 9: $h_2 \leadsto h_1$
lines 8,9: $h_3 \leadsto h_2$ and $h_1 \leadsto h_0$

Page 93 replacement for lines 13–15 ———— 12 Jun 1996

formula (6) and exercise 19 can be used; empirical tests were used to get approximate average values for the other cases. Ten thousand random files of 1000 elements were generated, and they each were sorted with each of the sequences of increments. The standard deviation of the number of left-to-right minima A was usually about 15; the standard deviation of the number of moves B was usually about 300.

Page 93 line 24 through page 95 line −3, major rewrite ———— 12 Oct 1996

the numbers $\bigl(2^k - (-1)^k\bigr)/3$ and $(3^k - 1)/2$, as well as Fibonacci numbers and the Incerpi-Sedgewick sequences (8) for $\rho = 2.5$ and $\rho = 2$. The final examples in Table 6 come from another sequence devised by Sedgewick, based on slightly different heuristics [*J. Algorithms* **7** (1986), 159–173]:

$$h_s = \begin{cases} 9 \cdot 2^s - 9 \cdot 2^{s/2} + 1, & \text{if } s \text{ is even;} \\ 8 \cdot 2^s - 6 \cdot 2^{(s+1)/2} + 1, & \text{if } s \text{ is odd.} \end{cases} \tag{11}$$

Table 6

ANALYSIS OF ALGORITHM D WHEN $N = 1000$

Increments													A_{ave}	B_{ave}	T
												1	6	249750	1
											17	1	65	41667	2
										60	6	1	158	26361	3
									140	20	4	1	262	21913	4
								256	64	16	4	1	362	20459	5
							576	192	48	16	4	1	419	20088	6
						729	243	81	27	9	3	1	378	18533	7
			512	256	128	64	32	16	8	4	2	1	493	16435	10
				500	250	125	62	31	15	7	3	1	516	7655	9
				501	251	125	63	31	15	7	3	1	558	7370	9
				511	255	127	63	31	15	7	3	1	559	7200	9
					255	127	63	31	15	7	3	1	436	7445	8
						127	63	31	15	7	3	1	299	8170	7
							63	31	15	7	3	1	190	9860	6
								31	15	7	3	1	114	13615	5
			513	257	129	65	33	17	9	5	3	1	561	6745	10
				257	129	65	33	17	9	5	3	1	440	6995	9
					129	65	33	17	9	5	3	1	304	7700	8
						65	33	17	9	5	3	1	197	9300	7
							33	17	9	5	3	1	122	12695	6
			683	341	171	85	43	21	11	5	3	1	511	7365	10
				341	171	85	43	21	11	5	3	1	490	7490	9
							255	63	15	7	3	1	373	8620	6
							257	65	17	5	3	1	375	8990	6
							341	85	21	5	3	1	410	9345	6
377	233	144	89	55	34	21	13	8	5	3	2	1	518	7400	13
	233	144	89	55	34	21	13	8	5	3	2	1	432	7610	12
						377	144	55	21	8	3	1	456	8795	7
						365	122	41	14	5	2	1	440	8085	7
							364	121	40	13	4	1	437	8900	6
								121	40	13	4	1	268	9790	5
						336	112	48	21	7	3	1	432	7840	7
				306	170	90	45	18	10	5	2	1	465	6755	9
					929	505	209	109	41	19	5	1	512	7725	8
						505	209	109	41	19	5	1	519	7790	7
							209	109	41	19	5	1	382	8165	6

When these increments $(h_0, h_1, h_2, \ldots) = (1, 5, 19, 41, 109, 209, \ldots)$ are used, Sedgewick proved that the worst-case running time is $O(N^{4/3})$.

The minimum number of moves, about 6750, was observed for increments of the form $2^k + 1$, and also in the Incerpi-Sedgewick sequence for $\rho = 2$. But it is important to realize that the number of moves is not the only consideration, even though it dominates the asymptotic running time. Since Program D takes $9B + 10NT + \cdots$ units of time, we see that saving one pass is about as desirable as saving $\frac{10}{9}N$ moves; when $N = 1000$ we are willing to add 1100 moves if we can save one pass. Therefore it seems unwise to start with h_{t-1} greater than, say, $\frac{1}{3}N$, since a large increment will not decrease the subsequent number of moves enough to justify the first pass.

Extensive empirical tests conducted by M. A. Weiss [*Comp. J.* **34** (1991), 88–91] suggest strongly that the average number of moves performed by Algorithm D with increments 2^k-1, …, 15, 7, 3, 1 is approximately proportional to $N^{5/4}$. More precisely, Weiss found that $B_{\rm ave} \approx 1.55N^{5/4} - 4.48N + O(N^{3/4})$ for $100 \le N \le 12000000$ when these increments are used; the empirical standard deviation was approximately $.065N^{5/4}$. He also discovered that Sedgewick's sequence (11) gives asymptotically better performance, $B_{\rm ave} \approx 0.43N^{7/6} + 18.5N + O(N^{5/6})$. Surprisingly, the standard deviation for this sequence of increments appears to be quite small, approximately of order $N^{3/4}$.

Table 7 shows typical breakdowns of moves per pass obtained in three random experiments, using increments of the forms 2^k-1, 2^k+1, and (11). The same file of numbers was used in each case. The total number of moves, $\sum_s B_s$, comes to 346152, 329532, 248788 in the three cases, so sequence (11) is clearly superior in this example.

Table 7

MOVES PER PASS: EXAMPLES WITH $N = 20000$

h_s	B_s	h_s	B_s	h_s	B_s
4095	19458	4097	19459	3905	20714
2047	15201	2049	14852	2161	13428
1023	16363	1025	15966	929	18206
511	18867	513	18434	505	16444
255	23232	257	22746	209	21405
127	28034	129	27595	109	19605
63	33606	65	34528	41	26604
31	40350	33	45497	19	23441
15	66037	17	48717	5	38941
7	43915	9	38560	1	50000
3	24191	5	20271		
1	16898	3	9448		
		1	13459		

Although Algorithm D is gradually becoming better understood, more than three decades of research have failed to turn up any grounds for making strong assertions about what sequences of increments make it work best. If N is less than 1000, a simple rule such as

$$\textit{Let } h_0 = 1,\ h_{s+1} = 2h_s + 1, \textit{ and stop with } h_{t-1} \textit{ when } 3h_t \ge N \qquad (12)$$

seems to be about as good as any other. For larger values of N, Sedgewick's sequence (11) can be recommended, again stopping with h_{t-1} when $3h_t \ge N$.

Page 96 line −14 through page 104 line −1 ——— 12 Oct 1996

Here we must renumber equations (9) through (15), calling them (13) through (19) instead. (8 changes)

▸**Page 99** line 21 ——— 30 Apr 1996

Kronmal and Tartar $\leadsto$ Tarter and Kronmal

▸**Page 102** line 4 ——— 26 Oct 1997

$4M-3 \leadsto 4M-3\delta$

Page 103 change in notation, continued —— 28 May 1995

line −7: $h_s = 2^{s-1}$ for $t \geq s \geq 1 \leadsto h_s = 2^s$ for $t > s \geq 0$
lines −3 through −1: decrease all subscripts on h by 1 (8 changes)

Page 104 rating of exercise 20 —— 30 Apr 1996

[M20] $\leadsto$ *[M22]*

Page 104 replacement for exercise 21 —— 30 May 1995

21. *[M25]* Let h and k be relatively prime positive integers, and say that an integer is *generable* if it equals $xh + yk$ for some nonnegative integers x and y. Show that n is generable if and only if $hk - h - k - n$ is not generable. (Since 0 is the smallest generable integer, the largest nongenerable integer must therefore be $hk - h - k$. It follows that $K_i \leq K_j$ whenever $j - i \geq (h-1)(k-1)$, in any file that is both h-ordered and k-ordered.)

Page 104 replacement for exercise 27 —— 29 May 1995

27. *[M41]* (Bjorn Poonen.) (a) Prove that there is a constant c such that if m of the increments h_s in Algorithm D are less than $N/2$, the running time is $\Omega(N^{1+c/\sqrt{m}})$ in the worst case. (b) Consequently the worst case running time is $\Omega\bigl(N(\log N/\log\log N)^2\bigr)$ for all sequences of increments.

Page 104 replacement for exercise 29 —— 12 Oct 1996

29. *[40]* For N=1000 and various values of t, find empirical values of $h_{t-1}, \ldots, h_1, h_0$ for which the average number of moves, B_{ave}, is as small as you can make it.

Page 104 line −2 —— 14 Jun 1977

inversions. $\leadsto$ inversions. Discuss corresponding improvements to Program S.

Page 105 replacement for exercise 39, which moved to page 72 —— 28 May 1995

39. *[HM26]* (a) Prove that the sum of the first $\binom{k}{2}$ elements of (10) is $O(\rho^{2k})$. (b) Now prove Theorem I.

Page 105 replacement for exercise 40 —— 30 May 1995

40. *[HM43]* Analyze the average behavior of Algorithm D when there are $t = 3$ increments h, g, and 1, assuming that $h \perp g$. The first pass, h-sorting, obviously does a total of $\frac{1}{4}N^2/h + O(N)$ moves. (a) Prove that the second pass, g-sorting, does $\frac{\sqrt{\pi}}{8}(\sqrt{h} - 1/\sqrt{h})N/g + O(hN)$ moves. (b) Prove that the third pass, 1-sorting, does $\psi(h,g)N + O(g^3h^2)$ moves, where

$$\psi(h,g) = \frac{1}{2}\sum_{d=1}^{g-1}\sum_j \binom{h-1}{j}\left(\frac{d}{g}\right)^j\left(1-\frac{d}{g}\right)^{h-1-j}\left|j - \left\lfloor\frac{hd}{g}\right\rfloor\right|.$$

Page 107 append new sentence at the end of step B2 —— 14 May 1976

(If `BOUND` = 1, this means go directly to B4.)

▸**Page 108** line −14 —— 07 Feb 1979

betweeen $\leadsto$ between

Page 117 eliminating unnecessary superfluous redundancy ——— 19 Dec 1976

line 5: $K \leftarrow K_l$, $R \leftarrow R_l$. $\leadsto$ $K \leftarrow K_l$.
line 6: K and R $\leadsto$ K

Page 118 program line *05* ——— 19 Dec 1976

$K \leftarrow K_l$, $R \leftarrow R_l$. $\leadsto$ $K \leftarrow K_l$.

▸**Page 118** program line *15* ——— 06 Jan 1997

$j \to j-1$. $\leadsto$ $j \leftarrow j-1$.

Page 120 line 9 ——— 14 May 1976

(December, 1974). $\leadsto$ (1974), 287–289.

Page 121 line −2 ——— 05 May 1996

In Section 6.2.2, we shall prove that $\leadsto$ Exercise 6.2.2–8 proves that, when $M=1$,

Page 122 line −6 ——— 19 Dec 1976

instructions "$K \leftarrow K_l$, $R \leftarrow R_l$" $\leadsto$ instruction "$K \leftarrow K_l$"

Page 123 new paragraph to follow line 22 ——— 27 May 1995

Robert Sedgewick has analyzed a number of optimized variants of quicksort in *Acta Informatica* **7** (1977), 327–356, and in *CACM* **21** (1978), 847–857, **22** (1979), 368. See J. L. Bentley and M. D. McIlroy, *Software Practice & Exp.* **23** (1993), 1249–1265, for a tuned-up version of quicksort based on 15 further years of experience.

▸**Page 127** program line *38* ——— 07 Jan 1997

$l \to l+1$. $\leadsto$ $l \leftarrow l+1$.

Page 128 line −3 ——— 19 Apr 1977

v. $\leadsto$ v. Yihsiao Wang has suggested that the mean of three key values such as (28) be used as the threshold for partitioning; he has proved that the number of comparisons required to sort uniformly distributed random data will then be asymptotic to $1.082N \lg N$.

▸**Page 131** in (37) ——— 19 Feb 1986

$\frac{31}{36}$ $\leadsto$ $\frac{49}{36}$

▸**Page 132** middle of page ——— 27 May 1978

7 lines after (42): $O(N^{t-1/2}e^{-\pi N/2})$ $\leadsto$ $O\bigl(|t+iN|^{t-1/2}e^{-t-\pi N/2}\bigr)$
10 lines after (42): $(N/x)^t$ $\leadsto$ $(N/xe)^t$

Page 132 line 8 after (42) ——— 30 Sep 1996

A. Erdélyi et al., $\leadsto$ Erdélyi, Magnus, Oberhettinger, and Tricomi,

▸**Page 133** middle of page ——— 27 May 1978

3 lines after (45): N^t $\leadsto$ $|M+iN|^t$
4 lines after (45): negligible. $\leadsto$ negligible, when N and N' are much larger than M.

▸**Page 134** Eq. (46) and the line following ——— 28 Feb 1978

, ⇝ $+ O(n^{-M})$
where ⇝ for arbitrarily large M, where

Page 134 displayed formula in line 12 ——— 28 Feb 1978

$$\bigl|f(n)\bigr| < 0.000000173\,;$$

Page 134 future changes . . . . 23 May 1995

I will eventually revise pages 132–134 and the related exercises so that they provide a better introduction to the now-extensive literature of Mellin transforms; this will simplify parts of other analyses, especially in Section 6.3. I should also mention that $T_{2n}/2 = T_n + e^{-n} - 1 + n$; hence we can see for example that the error term $O(n^{-M})$ newly added to (46) is essential. I should also bring exercise 54 into the text, at least to mention that it sheds more light on $f(n)$ and gives an exact formula without the need for $O(n^{-M})$.

Page 135 rating for exercise 16 ——— 29 Nov 1977

[*HM46*] ⇝ [*HM42*]

Page 137 bottom . . . . 23 May 1995

Exercises 41 and 42 should eventually be replaced by something about a more interesting topic.

▸**Page 138** exercise 46 ——— 14 Jun 1977

$\int_{a+i\infty}^{a+i\infty}$ ⇝ $\int_{a-i\infty}^{a+i\infty}$

▸**Page 138** line 3 of exercise 51 ——— 18 Apr 1995

$r_k(m)$. (Cf. Eq. 35). This ⇝ the quantity $r_k(m)$ in (35). (This

▸**Page 138** exercise 52 ——— 14 Jun 1977

$\left(\dfrac{2n}{n+t}\right)$ ⇝ $\dbinom{2n}{n+t}$

Page 139 new exercise ——— 14 Sep 1996

58. [*HM25*] For any real number $c > 0$, prove that the probability is less than e^{-c} that Algorithm Q will make more than $(c + 1)(N + 1)H_N$ comparisons when sorting random data. (This upper bound is especially interesting when c is, say, N^ϵ.)

Page 139 line -3 ——— 21 May 1996

K_i. ⇝ K_i, where i is as large as possible.

▸**Page 148** line 3 ——— 21 Mar 1976

r15 ⇝ rI5

Page 149 top ——— 19 Oct 1996

line 2: $0.9N$ ⇝ $0.94N$
line 6: $0.2N$ ⇝ $0.01N$

Page 153 line 4 below the tree ——— 14 May 1976

if and only if one ⇝ if and only if at least one

Page 153 to precede line −3 . . . . 13 Sep 1996

Newer methods for priority queues need to be discussed too — binomial queues [J. Vuillemin, CACM **12** *(1978), 309–315; M. R. Brown, SICOMP* **7** *(1978), 298–319]; pagodas [J. Françon, G. Viennot, and J. Vuillemin, FOCS* **19** *(1978), 1–7]; pairing heaps [M. L. Fredman, R. Sedgewick, D. D. Sleator, and R. E. Tarjan, Algorithmica* **1** *(1986), 111–129]; skew heaps [D. D. Sleator and R. E. Tarjan, SICOMP* **15** *(1986), 52–59; Fibonacci heaps [M. L. Fredman and R. E. Tarjan, JACM* **34** *(1987), 596–615]; calendar queues [R. Brown, CACM* **31** *(1988), 1220–1227; G. A. Davison, CACM* **32** *(1989), 1241–1243]; relaxed heaps [J. R. Driscoll, H. N. Gabow, R. Shrairman, and R. E. Tarjan, CACM* **31** *(1988), 1343–1354]; Fishspear [M. J. Fischer and M. S. Paterson, JACM* **41** *(1994), 3–30]; etc. Not all of these methods will survive the test of time; leftist trees are in fact already obsolete. Detailed implementations of binomial queues and Fibonacci heaps can be found in D. E. Knuth, The Stanford GraphBase (New York: ACM Press, 1994), 475–489.*

Page 156 replacement for bottom two lines ——— 19 Oct 1996

$$C''_N \approx \tfrac{1}{2}N\lg N - 1.41N\,;$$
$$D''_N \approx \lg N \pm 2\,;$$

Page 157 replacement for lines 1–4 ——— 19 Oct 1996

but no adequate theoretical explanation for the behavior of D''_N or for the conjectured constants 0.152, 2.61, or 1.41 has yet been found. The leading terms $B''_N \sim N\lg N$ and $C''_N \sim \frac{1}{2}N\lg N$ have, however, been established in an elegant manner by R. Schaffer and R. Sedgewick; see exercise 30. Schaffer has also proved that the minimum and maximum possible values of C''_N are respectively asymptotic to $\frac{1}{4}N\lg N$ and $\frac{3}{4}N\lg N$.

▸**Page 157** line 4 of exercise 6 ——— 07 Mar 1984

n^2 ⇝ N^2 (twice)

Page 157 rating change for exercise 7 ——— 23 May 1995

[*M46*] ⇝ [*M41*]

▸**Page 158** line −5 ——— 27 Mar 1978

$a_i x^{i_1}$ ⇝ $a_1 x^{i_1}$

Page 158 replacement for exercise 30 ——— 19 Oct 1996

▸**30.** [*HM35*] (R. Schaffer and R. Sedgewick.) Let h_{nm} be the number of heaps on the elements $\{1, 2, \ldots, n\}$ for which the selection phase of heapsort does exactly m promotions. Prove that $h_{nm} \le 2^m \prod_{k=2}^{n} \lg k$, and use this relation to show that the average number of promotions performed by Algorithm H is $N\lg N + O(N\log\log N)$.

Page 159 exercise 34 ———— 15 Mar 1995

[*M47*] $\leadsto$ [*HM41*] $\leadsto$
[The $\leadsto$ The
38, ...; is there a simple asymptotic formula?] $\leadsto$ 38, 87, 203, 482, 1160, ...; show that the number is asymptotically $ab^N N^{-3/2}$ for suitable constants a and b, using techniques like those of exercise 2.3.4.4–4.

▸**Page 159** line 4 of exercise 35 ———— 24 Feb 1998

ancestors until $\leadsto$ ancestors, possibly swapping left and right subtrees, until

Page 159 new exercise ———— 19 Oct 1996

38. [*M28*] Prove that the number of heapsort promotions, B, is always at least $\frac{1}{2}N\lg N + O(N)$, if the keys are distinct. *Hint:* Consider the movement of the largest $\lceil N/2\rceil$ keys.

Page 159 new exercise ———— 29 Dec 1996

39. [*HM32*] Let $e_N(k)$ be the expected treewise distance of the kth-largest element from the root, in a random heap of N elements, and let $e(k) = \lim_{N\to\infty} e_N(k)$. Thus $e(1) = 0$, $e(2) = 1$, $e(3) = 1.5$, and $e(4) = 1.875$. Find the asymptotic value of $e(k)$ to within $O(k^{-1})$.

▸**Page 165** bottom line of Table 2 ———— 01 Jul 1975

179 $\leadsto$ 170

▸**Page 167** program line *21* ———— 27 May 1978

$p \leftarrow L_q$. $\leadsto$ $p \leftarrow L_p$.

▸**Page 176** line −12 ———— 27 May 1978

$M = b \leadsto M = b^r$

Page 177 top . . . . 23 May 1995

These comments will be simpler and sharper when `MMIX` replaces `MIX`.

Page 177 lines 25–27 ———— 21 Sep 1976

that the multiplicity ... Algorithm R, even $\leadsto$
that it ultimately spends too much time fussing with very small piles. Algorithm R is relatively efficient, even

▸**Page 178** line 6 ———— 05 Nov 1997

18.0 $\leadsto$ 18.1

Page 180 new exercise ———— 31 May 1995

▸**16.** [*25*] Design an algorithm to sort strings $\alpha_1, \ldots, \alpha_n$ on an m-letter alphabet into lexicographic order. The total running time of your algorithm should be $O(m+n+N)$, where $N = |\alpha_1| + \cdots + |\alpha_n|$ is the total length of all the strings.

Page 186 bottom line . . . . 17 Jun 1997

[Exercise 4.5.2–27 has become exercise 4.5.2–36 in the third edition of Volume 2.]

Page 188 line 10 ——— 21 Apr 1995

C. Ping $\leadsto$ P. Czen

Page 188 line 12 ——— 11 Jan 1997

Ping $\leadsto$ Czen

▸**Page 188** the binary value of 14! ——— 10 Nov 1997

1010001001100001110110010100000000000

Page 192 line -2 ——— 27 May 1978

Well's $\leadsto$ Wells's

Page 193 Fig. 36 . . . . 21 Oct 1997

I think there was a bug in my program that originally generated this figure. In the next edition I will present data in which I have more faith.

Page 193 lines -12 and -11 ——— 23 May 1995

But as yet nobody has discovered *any* n $\leadsto$ Glenn K. Manacher [*JACM* **26** (1979), 441–456] has proved that infinitely many $n \geq 189$ exist

Page 193 line -7 ——— 05 May 1996

method! $\leadsto$ method! At present, the smallest case where $F(n)$ is definitely known to be nonoptimum is $n = 47$: After sorting 5 and 42 elements with $F(5) + F(42) = 178$ comparisons, we can merge the results with 22 further comparisons, using a method due to J. S. Mönting, *Theoretical Comp. Sci.* **14** (1981), 19–37; this beats $F(47) = 201$.

Page 194 line -6 ——— 23 May 1995

unpublished memorandum $\leadsto$ unpublished IBM memorandum

Page 198 new exercise ——— 05 May 1996

35. [*M50*] (S. S. Kislitsyn, 1968.) Prove or disprove: Any directed graph G with $T(G) > 1$ has two vertices u and v such that the graphs G_1 and G_2 obtained from G by adding the arcs $u \leftarrow v$ and $u \rightarrow v$ satisfy $1 \leq T(G_1)/T(G_2) \leq 2$. (Thus $T(G_1)/T(G)$ always lies between $1/3$ and $2/3$, for some u and v.)

▸**Page 199** Eq. (4) ——— 28 Feb 1978

$$\lg \left\lceil \binom{m+n}{m} \right\rceil \leadsto \left\lceil \lg \binom{m+n}{m} \right\rceil$$

Page 200 revision of lines 7–23 ——— 14 May 1976

Constructing lower bounds. Theorem M shows that the "information theoretic" lower bound (2) can be arbitrarily far from the true value; thus the technique used to prove Theorem M gives us another way to discover lower bounds. Such a proof technique is often viewed as the creation of an *adversary*, a pernicious being who tries to make algorithms run slowly. When an algorithm for merging decides to compare $A_i : B_j$, the adversary determines the fate of the comparison so as to force the algorithm down the more difficult path. If we can invent a suitable adversary, as in the proof of Theorem M, we can ensure that every valid merging algorithm will have to make quite a few comparisons.

We shall make use of *constrained adversaries*, whose power is limited with regard to the outcomes of certain comparisons. A merging method that is under the influence of a constrained adversary does not know about the constraints, so it must make the necessary comparisons even though their outcomes have been predetermined. For example, in our proof of Theorem M we constrained all outcomes by condition (5), yet the merging algorithm was unable to make use of that fact in order to avoid any of the comparisons.

The constraints we shall use in the following discussion apply to the left and right ends of the files. Left constraints are symbolized by

Page 200 further changes ——— 14 May 1976

lines −13, −12, −9, −8: pronouncements ⇝ outcomes
lines −7, −4: oracles ⇝ adversaries
lines −6, −5, −3: oracle ⇝ adversary
line −2: wise, oracle ⇝ dangerous, adversary

Page 201 more of the same ——— 14 May 1976

lines 2, 9, 10, 18, 18: oracle ⇝ adversary
line 3: The oracle can use six strategies in general which ⇝ Six general strategies can be used to
lines 7, 16: questions A_p:B_q will be answered ⇝ comparisons $A_p : B_q$ will result in

Page 202 in the same vein ——— 14 May 1976

lines 2, 11, 16, 18, 20, −9,−6: oracle ⇝ adversary
line 12: then we define $\nabla M\phi(m,n)$ to be ⇝ thus, $\nabla M\phi(m,n)$ is
line 15: our oracle ⇝ that our adversary

Page 203 caption to Table 1 ——— 14 May 1976

"ORACLE" ⇝ ADVERSARY

Page 204 lines 1, 3, 5, 11 ——— 14 May 1976

oracle ⇝ adversary

Page 204 lines −12 and −11 ——— 24 Jun 1980

This proof ... 6.) ⇝ The reader may have noticed a pattern in the three formulas just proved. Paul Stockmeyer and Frances Yao have shown that the pattern holds in general, i.e., that the lower bounds derived by the strategy above suffice to establish the values $M(m, m+d) = 2m + d - 1$ for $m \geq 2d - 2$. [*SICOMP* **9** (1980), 85–90.]

Page 206 line 2 before (22) ——— 05 May 1996

exercise 16). ⇝ exercise 16). Improved algorithms have been found by F. K. Hwang and D. N. Deutsch, *JACM* **20** (1973), 148–159; G. K. Manacher, *JACM*

26 (1979), 434–440; and most notably by C. Christen, *FOCS* **19** (1978), 259–266. Christen's merging procedure, called *forward-testing-backward-insertion*, saves about $m/3$ comparisons over Algorithm H when $n/m \to \infty$. It achieves the lower bound $.M.(m,n) = \lfloor (11m+n-3)/4 \rfloor$ when $5m-3 \le n \le 7m+2[m \text{ even}]$; hence it is optimum in such cases (and, remarkably, so is our adversarial lower bound).

Page 207 new rating for exercise 4 ——— 07 May 1996

[*M44*] $\leadsto$ [*M42*]

Page 207 exercise 10 line 2 ——— 14 May 1976

oracle $\leadsto$ adversary

▸**Page 207** line 5 of exercise 11 ——— 13 Jan 1997

$n \ge g_{t-1} \leadsto n = g_t$ or $n = g_t - 1$

Page 208 replacement for exercise 14 ——— 29 Nov 1977

14. [*41*] (F. K. Hwang.) Let $h_{3k} = \lfloor (43/28)\cdot 2^k \rfloor - 1$, $h_{3k+1} = h_{3k} + 3\cdot 2^{k-3}$, $h_{3k+2} = \lfloor (17/7)\cdot 2^k - 6/7 \rfloor$ for $k \ge 3$, and let the initial values be defined so that

$$(h_0, h_1, h_2, \ldots) = (1, 1, 2, 2, 3, 4, 5, 7, 9, 11, 14, 18, 23, 29, 38, 48, 60, 76, \ldots).$$

Prove that $M(3, h_t) > t$ and $M(3, h_t - 1) \le t$ for all t, thereby establishing the exact values of $M(3,n)$ for all n.

Page 209 rating change for exercise 22 ——— 23 May 1995

[*M50*] $\leadsto$ [*M43*]

Page 209 exercise 23 line 6 ——— 14 May 1976

oracle $\leadsto$ adversary

Page 209 new exercise ——— 17 Oct 1996

▸**24.** [*25*] Suppose (x_{ij}) is an $m \times n$ matrix with nondecreasing rows and columns: $x_{ij} \le x_{(i+1)j}$ for $1 \le i < m$ and $x_{ij} \le x_{i(j+1)}$ for $1 \le j < n$. Show that $M(m,n)$ is the minimum number of comparisons needed to determine whether a given number x is present in the matrix, if all comparisons are between x and some matrix element.

Page 211 line 21 ——— 06 May 1996

Slupecki $\leadsto$ Słupecki

Page 211 line −3 ——— 14 May 1976

"oracle," $\leadsto$ adversary as in Section 5.3.2,

Page 212 in same vein ——— 14 May 1976

line 12: finding an oracle $\leadsto$ constructing an adversary
lines 14, 16, 19, 22: oracle $\leadsto$ adversary

Page 215 replacement for the 9 lines preceding Table 1 ———— 06 May 1996

no general method is evident as yet. The values listed for $V_4(9) = V_6(9)$ and $V_5(10) = V_6(10)$ were proved optimum in 1996 by W. Gasarch, W. Kelly, and W. Pugh [*SIGACT News* **27**, 2 (June 1996), 88–96], using a computer search.

A fairly good lower bound for the selection problem when t is small was obtained by David G. Kirkpatrick [*JACM* **28** (1981), 150–165]: If $2 \le t \le (n+1)/2$, we have

$$V_t(n) \ge n + t - 3 + \sum_{j=0}^{t-2} \left\lceil \lg \frac{n-t+2}{t+j} \right\rceil . \qquad (12)$$

In his Ph.D. thesis [U. of Toronto, 1974], Kirkpatrick proved that

$$V_3(n) \le n + 1 + \left\lceil \lg \frac{n-1}{4} \right\rceil + \left\lceil \lg \frac{n-1}{5} \right\rceil ; \qquad (13)$$

this upper bound matches the lower bound (12) for $\lg \frac{5}{3} \approx 74\%$ of all integers n, and it exceeds (12) by at most 1. Kirkpatrick's analysis made it natural to conjecture that equality holds in (13) for all $n > 4$, but Jutta Eusterbrock found the surprising counterexample $V_3(22) = 28$ [*Discrete Applied Math.* **41** (1993), 131–137]. Improved lower bounds for larger values of t were found by S. W. Bent and J. W. John (exercise 26):

$$V_t(n) \ge n + m - 2\lceil \sqrt{m}\, \rceil, \qquad m = 2 + \left\lceil \lg \left(\binom{n}{t} \Big/ (n+1-t) \right) \right\rceil . \qquad (14)$$

This formula proves in particular that

$$V_{\alpha n}(n) \ge \left(1 + \alpha \lg \frac{1}{\alpha} + (1-\alpha) \lg \frac{1}{1-\alpha} \right) n + O(\sqrt{n}\,) . \qquad (15)$$

Page 215 new caption for Table 1 ———— 06 May 1996

VALUES OF $V_t(n)$ FOR SMALL n

Page 215 bottom line of Table 1, and an additional footnote ———— 02 Mar 1977

17 $\rightsquigarrow$ 16** (twice)

** See K. Noshita, *Trans. of the IECE of Japan* **E59**, 12 (Dec. 1976), 17–18.

Page 217 lines 17–20 ———— 19 Dec 1976

A. Schönhage ... at least $1.75n$. $\rightsquigarrow$ A. Schönhage, M. Paterson, and N. Pippenger [*J. Comp. Sys. Sci.* **13** (1976), 184–199] proved that the maximum number of comparisons required to find the median is at most $3n + O(n \log n)^{3/4}$. See exercise 27 for a lower bound and for references to more recent results.

Page 217 line 27 ———— 07 May 1996

1968)]. $\rightsquigarrow$ 1968); *Revue Française d'Automatique, Informatique et Recherche Opérationnelle* **6**, R-3 (December 1972), 23–68].

Page 217 replacement for the bottom three lines ———— 11 Sep 1996

with only $\frac{3}{2}n + O(n^{2/3}\log n)$ comparisons, on the average. He and R. L. Rivest refined this method a few years later and constructed an elegant algorithm to prove that

$$\overline{V}_t(n) \le n + \min(t, n-t) + O(\sqrt{n\log n}\,). \qquad (16)$$

(See exercises 13 and 24.)

Page 218 update the paragraph before the exercises ———— 11 Feb 1995

[to appear] $\leadsto$
(13) $\leadsto$ (17)
No really satisfactory ... have yet been obtained. $\leadsto$ An elegant lower bound on $\overline{V}_t(n)$ appears in exercise 25.

Page 218 new paragraph before the exercises ———— 02 Sep 1996

The sorting and selection problems are special cases of the much more general problem of finding a permutation of n given elements that is consistent with a given partial ordering. A. C. Yao [*SICOMP* **18** (1989), 679–689] has shown that, if the partial ordering is defined by an acyclic digraph G on n vertices with k connected components, the minimum number of comparisons necessary to solve such problems is always $\Theta\bigl(\lg\bigl(n!/T(G)\bigr) + n - k\bigr)$, in both the worst case and on the average, where $T(G)$ is the total number of permutations consistent with the partial ordering (i.e., the number of topological sortings of G).

Page 219 replacements for exercises 3 and 4 ———— 25 Aug 1996

3. [*20*] Prove that $V_t(n) > V_t(n-1)$ and $W_t(n) > W_t(n-1)$, for $1 \le t < n$.

▸**4.** [*M25*] (F. Fussenegger and H. N. Gabow.) Prove that $W_t(n) \ge n - t + \lceil \lg n^{\underline{t-1}} \rceil$.

Page 219 new wording of exercise 14 ———— 19 Dec 1976

▸**14.** [*20*] (M. Sobel.) Let $U_t(n)$ be the minimum number of comparisons needed to find the t largest of n elements, without necessarily knowing their relative order. Show that $U_2(5) \le 5$.

Page 220 lines 5–6 ———— 14 May 1976

a suitable oracle. $\leadsto$ an adversary.

Page 220 replacement for exercise 21 ———— 25 Aug 1996

21. [*25*] (P. V. Ramanan and L. Hyafil.) Prove that $W_t(2^k + 2^{k+1-t}) \le 2^k + 2^{k+1-t} + (t-1)(k-1)$, when $k \ge t \ge 2$; also show that equality holds for infinitely many k and t, because of exercise 4. [*Hint:* Maintain two knockout trees and merge their results cleverly.]

Page 220 new version of exercise 22 ———— 14 May 1976

22. [*24*] (David G. Kirkpatrick.) Show that when $4 \cdot 2^k < n - 1 \le 5 \cdot 2^k$, the upper bound (11) for $V_3(n)$ can be reduced by 1 as follows: (i) Form four "knockout trees" of size 2^k. (ii) Find the minimum of the four maxima, and discard all 2^k elements of its tree. (iii) Using the known information, build a single knockout tree of size $n - 1 - 2^k$. (iv) Continue as in the proof of (11).

Page 220 new version of exercise 24 ______ 13 Sep 1996

24. [*HM40*] Prove that $\overline{V}_t(n) \le n + t + O(\sqrt{n \log n}\,)$ for $t \le \lceil n/2 \rceil$. *Hint:* Show that with this many comparisons we can in fact find both the $\lfloor t - \sqrt{t \ln n}\,\rfloor$th and $\lceil t + \sqrt{t \ln n}\,\rceil$th elements, after which the tth is easily located.

Page 220 new exercise ______ 11 Sep 1996

▸**25.** [*M35*] (W. Cunto and J. I. Munro.) Prove that $\overline{V}_t(n) \ge n + t - 2$ when $t \le \lceil n/2 \rceil$.

Page 220 new exercise ______ 19 Dec 1976

26. [*M32*] (A. Schönhage, 1974.) (a) In the notation of exercise 14, prove that $U_t(n) \ge \min\bigl(2 + U_t(n-1), 2 + U_{t-1}(n-1)\bigr)$ for $n \ge 3$. [*Hint:* Construct an adversary by reducing from n to $n-1$ as soon as the current partial ordering is not composed entirely of components having the form • or •—•.] (b) Similarly, prove that

$$U_t(n) \ge \min\bigl(2 + U_t(n-1), 3 + U_{t-1}(n-1), 3 + U_t(n-2)\bigr)$$

for $n \ge 5$, by constructing an adversary that deals with components •, •—•, , . (c) Therefore we have $U_t(n) \ge n + t + \min\bigl(\lfloor (n-t)/2 \rfloor, t\bigr) - 3$ for $1 \le t \le n/2$. [The inequalities in (a) and (b) apply also when V or W replaces U, thereby establishing the optimality of several entries in Table 1.]

Page 220 new exercise ______ 14 Feb 1995

▸**27.** [*M34*] A *randomized adversary* is an adversary algorithm that is allowed to flip coins as it makes decisions. (a) Let A be a randomized adversary and let $\Pr(l)$ be the probability that A reaches leaf l of a given comparison tree. Show that if $\Pr(l) \le p$ for all l, the height of the comparison tree is $\ge \lg(1/p)$. (b) Consider the following adversary for the problem of selecting the tth largest of n elements, given integer parameters q and r to be selected later:

A1. Choose a random set T of t elements; all $\binom{n}{t}$ possibilities are equally likely. (We will ensure that the $t-1$ largest elements belong to T.) Let $S = \{1, \ldots, n\} \setminus T$ be the other elements, and set $S_0 \leftarrow S$, $T_0 \leftarrow T$; S_0 and T_0 will represent elements that might become the tth largest.

A2. While $|T_0| > r$, decide all comparisons $x : y$ as follows: If $x \in S$ and $y \in T$, say that $x < y$. If $x \in S$ and $y \in S$, flip a coin to decide, and remove the smaller element from S_0 if it was in S_0. If $x \in T$ and $y \in T$, flip a coin to decide, and remove the larger element from T_0 if it was in T_0.

A3. As soon as $|T_0| = r$, partition the elements into three classes P, Q, R as follows: If $|S_0| < q$, let $P = S$, $Q = T_0$, $R = T \setminus T_0$. Otherwise, for each $y \in T_0$, let $C(y)$ be the elements of S already compared with y, and choose y_0 so that $|C(y_0)|$ is minimum. Let $P = (S \setminus S_0) \cup C(y_0)$, $Q = (S_0 \setminus C(y_0)) \cup \{y_0\}$, $R = T \setminus \{y_0\}$. Decide all future comparisons $x : y$ by saying that elements of P are less than elements of Q are less than elements of R; flip a coin when x and y are in the same class.

Prove that if $1 \le r \le t$ and if $|C(y_0)| \le q - r$ at the beginning of step A3, each leaf is reached with probability $\le (n+1-t)/\bigl(2^{n-q}\binom{n}{t}\bigr)$. *Hint:* Show that at least $n - q$ coin flips are made. (c) Continuing (b), show that we have

$$V_t(n) \ge \min\Bigl(n - 1 + (r-1)(q+1-r),\; n - q + \lg\bigl(\tbinom{n}{t}/(n+1-t)\bigr)\Bigr),$$

for all integers q and r. (d) Establish (14) by choosing q and r.

Page 220 modifications to terminology ——— 14 May 1976

line −21: a *homogeneous* $\leadsto$ an *oblivious*
line −2: a homogeneous $\leadsto$ an oblivious
lines −2 and −1: any homogeneous $\leadsto$ any oblivious
caption on page 221: A homogeneous $\leadsto$ An oblivious

▸**Page 225** line 1 ——— 27 May 1978

$x_{\lfloor m/2 \rfloor} \leadsto x_{2\lfloor m/2 \rfloor}$
$y_{\lfloor n/2 \rfloor} \leadsto y_{2\lfloor n/2 \rfloor}$

Page 227 third line of (11) ——— 08 May 1996

46 $\leadsto$ 45

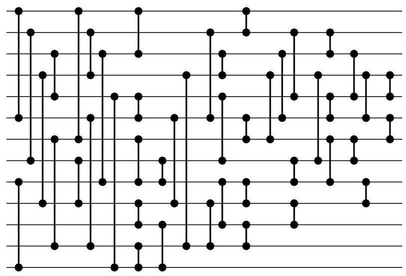
Page 228 line 8 ——— 14 May 1976

(North-Holland, 1972), Chapter 15 $\leadsto$ (North-Holland, 1973), 163–172

Page 228 new paragraph to follow line 11 ——— 08 May 1996

The network shown for $n = 13$ has quite a different pedigree: Hughes Juillé used a computer program to construct it in 1995, by simulating an evolutionary process of genetic breeding. The network exhibits no obvious rhyme or reason, but it works — and it's shorter than any other construction devised so far by human ratiocination.

Page 228 additional diagram for Fig. 49 ——— 08 May 1996



$n = 13$ 45 modules, delay 10

Page 229 line 16 ——— 08 May 1996

11, 13, 14, $\leadsto$ 11, 14,

Page 229 replacement for lines 18–22 ——— 23 May 1995

The best sorting network currently known for 256 elements, due to D. Van Voorhis, shows that $\hat{S}(256) \le 3651$, compared to 3839 by Batcher's method. [See R. L. Drysdale and F. H. Young, *SICOMP* **4** (1975), 264–270.] As $n \to \infty$, it turns out in fact that $\hat{S}(n) = O(n \log n)$; this astonishing upper bound was proved by Ajtai, Komlós, and Szemerédi, *Combinatorica* **3** (1983), 1–19. The networks they constructed are not of practical interest, since they use many comparators just to save a factor of $\log n$. Batcher's method is much better, unless n exceeds the total memory capacity of all computers on earth! But their theorem establishes the true asymptotic growth of $\hat{S}(n)$, up to a constant factor.

Page 230 line 1 after (14) ——— 23 May 1995

$n \le 8$ ⇝ $n \le 10$

Page 231 line −1 through page 232 line 3 ——— 14 Sep 1996

in general. The ... exercise 9.) ⇝ in general.

The odd-even merge shows that

$$\widehat{M}(m,n) \le C(m,n) = \tfrac{1}{2}(m+n)\lg\min(m,n) + O(m+n).$$

P. B. Miltersen, M. Paterson, and J. Tarui [*JACM* **43** (1996), 147–165] have improved Theorem F by establishing the lower bound

$$\widehat{M}(m,n) \ge \tfrac{1}{2}\bigl((m+n)\lg(m+1) - m/\ln 2\bigr) \qquad \text{for } 1 \le m \le n.$$

Consequently $\widehat{M}(m,n) = \frac{1}{2}(m+n)\lg\min(m,n) + O(m+n)$.

The exact formula $\widehat{M}(2,n) = C(2,n) = \lceil \frac{3}{2}n \rceil$ has been proved by A. C. Yao and F. F. Yao [*JACM* **23** (1976), 566–571]. The value of $\widehat{M}(m,n)$ is also known to equal $C(m,n)$ for $m = n \le 5$; see exercise 9.

Page 233 lines 1–8 ——— 07 Feb 1996

When $m = n$ ... in this case. Batcher ⇝ Bitonic merging is optimum, in the sense that no parallel merging method based on simultaneous disjoint comparisons can sort in fewer than $\lceil \lg(m{+}n) \rceil$ stages. (See exercise 46.) Another way to achieve this optimum time, with fewer comparisons but a slightly more complicated control logic, is discussed in exercise 49.

When $1 \le m \le n$, the nth smallest output of an (m,n) merging network depends on $2m + [m<n]$ of the inputs (cf. exercise 29). If it can be computed by comparators with l levels of delay, it involves at most 2^l of the inputs; hence $2^l \ge 2m + [m<n]$, and $l \ge \lceil \lg(2m + [m<n]) \rceil$. Batcher

Page 235 just before Table 1 ——— 23 May 1995

[To appear.] ⇝ [Ph.D. thesis, U. of Illinois, 1975.] N. Pippenger [*SICOMP* **20** (1991), 878–887] has proved by nonconstructive methods that for any $\epsilon > 0$ there exist selection networks with $\widehat{U}_{\lceil n/2 \rceil}(n) \le (2+\epsilon)n\lg n$, whenever n is sufficiently large (depending on ϵ).

▸**Page 237** line −2 ——— 11 Nov 1985

step 8 ... 2^q "−" ⇝ step q consist alternately of 2^{q-1} "+" followed by 2^{q-1} "−"

Page 243 rating changes ——— 23 May 1995

exercise 44: [*46*] ⇝ [*42*]
exercise 47: [*48*] ⇝ [*43*]
exercise 48: [*HM49*] ⇝ [*HM46*]

Page 243 replacement for exercise 46 ——— 07 Feb 1996

▸**46.** [*30*] (M. Aigner.) Show that the minimum number of stages needed to merge m elements with n, using simultaneous disjoint comparisons as in exercise 6, is $\lceil \lg(m+n) \rceil$; hence the bitonic merging network has optimum delay.

Page 243 replacement for exercise 49 ———————— 31 Oct 1996 bubble sort

49. [*M35*] The *even-odd merge* is similar to Batcher's odd-even merge, except that when $mn > 2$ it recursively merges the sequence $\langle x_{m \bmod 2+1}, \ldots, x_{m-3}, x_{m-1}\rangle$ with $\langle y_1, y_3, \ldots, y_{2\lceil n/2\rceil-1}\rangle$ and $\langle x_{(m+1) \bmod 2+1}, \ldots, x_{m-2}, x_m\rangle$ with $\langle y_2, y_4, \ldots, y_{2\lfloor n/2\rfloor}\rangle$ before making a set of $\lceil m/2\rceil + \lceil n/2\rceil - 1$ comparison-interchanges analogous to (1). Show that the even-odd merge achieves the optimum delay time $\lceil \lg(m+n)\rceil$ of bitonic merging, without making more comparisons than the bitonic method. In fact, prove that the number of comparisons $A(m,n)$ made by even-odd merging satisfies $C(m,n) \le A(m,n) < \frac{1}{2}(m+n)\lg\min(m,n) + m + \frac{3}{2}n$.

Page 243 replacement for exercise 51 ———————— 29 Jul 1996

▸**51.** [*25*] Suppose $x \in D_n$ is not sorted. Show that there is an n-network α_x that fails to sort x, although it sorts all other elements of D_n.

Page 246 new exercise ———————— 28 Apr 1996

63. [*25*] Suppose a primitive comparator network on n lines is known to sort the single input 1 0 1 0 ... 1 0 correctly. (Cf. exercise 36; assume that n is even.) Show that its "middle third," consisting of all comparators that involve only lines $\lceil n/3\rceil$ through $\lceil 2n/3\rceil$ inclusive, will sort *all* inputs.

Page 246 new exercises ———————— 16 May 1996

64. [*HM42*] Comparators $[i_1 : i_1+1][i_2 : i_2+1]\ldots[i_r : i_r+1]$ are chosen at random, with each value of $i_k \in \{1, 2, \ldots, n-1\}$ equally likely; the process stops when the network contains a bubble sort configuration like that of Fig. 47 as a subnetwork. Prove that $r \le 4n^2 + \sqrt{n}\,\lg n$, except with probability $O(n^{-1000})$.

65. [*M47*] Comparators $[i_1 : j_1][i_2 : j_2]\ldots[i_r : j_r]$ are chosen at random, with each *irredundant* choice $1 \le i_k < j_k \le n$ equally likely; the process stops when a sorting network has been obtained. What is the expected value of r? Is it, for example, $O(n^{1+\epsilon})$ for all $\epsilon > 0$?

Page 246 new exercise ———————— 23 May 1995

▸**66.** [*43*] Let $N = \binom{n}{2}$. Find a one-to-one correspondence between Young tableaux of shape $(n-1, n-2, \ldots, 1)$ and primitive sorting networks $[i_1 : i_1+1]\ldots[i_N : i_N+1]$. [Consequently by Theorem 5.1.4H there are exactly

$$\frac{N!}{1^{n-1}\,3^{n-2}\,5^{n-3}\ldots(2n-3)^1}$$

such sorting networks.] *Hint:* Exercise 36(c) shows that primitive networks without redundant comparators correspond to paths from $1\,2\ldots n$ to $n\ldots 2\,1$ in polyhedra like Fig. 1 in Section 5.1.1.

Page 246 new exercise (formerly exercise 49) ———————— 23 May 1995

67. [*46*] Is the function $T(n)$ of exercise 6 strictly less than $\widehat{T}(n)$ for some n?

Page 259 lines 4–7 ———————— 21 Sep 1976

has not yet ... $m = 8$. This increase
is difficult to analyze precisely, but T. O. Espelid has shown how to extend the snowplow analogy to obtain an approximate formula for the behavior [*BIT* **16**

(1976), 133–142]. According to his formula, which agrees well with empirical tests, the run length will be about

$$2P + (m - 1.5)\left(\frac{2P + (m-2)b}{2P + (2m-3)b}\right) b,$$

when b is the block size and $m \ge 2$. Such an increase

Page 259 line 19 ———— 17 Jan 1997

next string $\rightsquigarrow$ next run

Page 260 new paragraph to be inserted just before Table 2 ———— 28 Feb 1978

The ideas of delayed run-reconstitution and natural selection can be combined, as discussed by T. C. Ting and Y. W. Wang in *Comp. J.* **20** (1977), 298–301.

▸**Page 260** line −7 ———— 21 Nov 1997

$4.69445P \rightsquigarrow 4.69446P$

▸**Page 262** line 7 ———— 27 May 1978

$(4e - 10)P \rightsquigarrow \sqrt{(4e-10)P}$

▸**Page 264** line −7 ———— 27 May 1978

beings $\rightsquigarrow$ begins

Page 265 line 10 ———— 14 May 1976

[*Hint:* See ... Theorem 4.5.3L.] $\rightsquigarrow$

▸**Page 266** line 6 ———— 01 Jul 1975

other P. $\rightsquigarrow$ other P.]

Page 267 and following . . . . 23 May 1995

The material on tape sorting in Sections 5.4.2–5.4.8 will be substantially shortened in order to make room for other material that is now of much greater relevance.

Page 279 line 10 after Table 4 ———— 14 Jun 1977

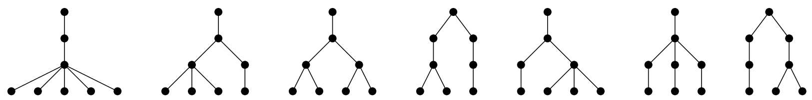
JACM (to appear) $\rightsquigarrow$ *SICOMP* **6** (1977), 1–39

▸**Page 284** top ———— 05 Jan 1979

lines 4, 13, 20: 25 $\rightsquigarrow$ 27
line 22: $143r \rightsquigarrow 145r$

Page 289 new exercise ———— 05 Nov 1996

31. [*M22*] (R. Kemp.) Let $K_d(n)$ be the number of n-node ordered trees in which every leaf is at distance d from the root. For example, $K_3(8) = 7$ because of the trees



Show that $K_d(n)$ is a generalized Fibonacci number, and find a one-to-one correspondence between such trees and the ordered partitions considered in exercise 8.

Page 293 line −11 ——— 21 May 1996

M[j], $1 \le j \le T$: ⇝ M[j], $1 \le j < T$:

▸**Page 294** line 15 ——— 07 Feb 1975

to C5. ⇝ to C5 if $m > 0$.

Page 294 line 2 of step C7 ——— 21 May 1996

`A, AA` ⇝ `AA`

▸**Page 312** line 2 of exercise 16 ——— 25 Nov 1997

(modulo $T-1$) ⇝ (modulo $T-2$)

▸**Page 317** correction to step B1 ——— 14 Nov 1979

transpose the two sentences 'Then write ...' ↔ 'Set $A[0,0]$ ...'

Page 318 line −10 ——— 17 Jan 1997

initial strings ⇝ initial runs

Page 319 lines 4–6 ——— 21 May 1996

line 2: Thus ⇝ Alas
lines 4–6: Since programming ... inevitable. ⇝ Fortunately the oscillating sort isn't especially good; let's hope that good guys who invent the best algorithms continue to make their ideas freely available.

▸**Page 326** line −7 ——— 29 Nov 1977

$L[p]$ ⇝ $L[m]$

▸**Page 336** line −15 ——— 23 Nov 1997

Table 5.4.2–5 ⇝ Table 5.4.2–6

Page 341 the foldout illustration ——— 31 Jul 1976

In the bottom example (10) look at line 4 of the six lines: There's a longish black bar as the seventh activity, about halfway between 0 minutes and 1 minute; the sixth activity in that line is a shorter black bar. Lines 1, 2, 3, and 5 have a blank bar just above and below the longish black bar; they should actually have parallel upward-slanting diagonal lines (the symbol for "reading in forward direction") inside their blank bars.

Page 342 bottom ——— 21 May 1996

line −6: `NOV201917` ⇝ `NOV071917`
line −2: `19171120` ⇝ `19171107`

▸**Page 345** lines 9 and 10 ——— 07 Dec 1997

.127 ⇝ .125

▸**Page 348** line 9 after the first illustration ——— 27 May 1978

tape C, 3's and 5's on tape D ⇝ tape A, 3's and 5's on tape B

▸**Page 351** line −14 ——— 04 Jul 1981

This is perfect ⇝ This is a perfect

▸**Page 352** exercise 3 ——— 29 Nov 1977

polyphase merge $\leadsto$ polyphase radix sort

▸**Page 352** line −9 ——— 14 Jun 1977

output them is $\leadsto$ output them in

▸**Page 352** line −4 ——— 30 Sep 1982

moment's $\leadsto$ moments'

▸**Page 354** in (3) ——— 14 Feb 1997

[The $\sum$ signs are too high.]

▸**Page 356** line −11 ——— 27 May 1978

T3 $\leadsto$ Track 3

▸**Page 358** line −20 ——— 19 Dec 1976

artifically $\leadsto$ artificially

▸**Page 370** lines 10 and 11 ——— 06 Jul 1981

Software Age $\leadsto$ *Software Age*

▸**Page 370** Equation (8) ——— 25 Aug 1976

$B_2^2 \leadsto B_1^2$

▸**Page 372** line 5 ——— 26 Dec 1997

$\min\limits_{1\le k\le n} \leadsto \min\limits_{1\le k<n}$

▸**Page 376** first display of exercise 2 ——— 12 Feb 1997

$a_{p1} \leadsto a_{P1}$
$a_{p2} \leadsto a_{P2}$
$a_{pL} \leadsto a_{PL}$

Page 378 new exercise ——— 10 Oct 1980

19. [*HM25*] (R. W. Floyd.) Show that the lower bound of Theorem F can be improved to

$$\frac{(k+1)nc\lg c + nc/\ln 2}{b+c}\left(1+O\left(\frac{\log c}{c}\right)\right)$$

when $n=c^k$, for fixed k as $c\to\infty$, and also to $nc+O(n/\log n)$ for fixed c as $n\to\infty$, in the sense that some initial configuration must require this many stops. [*Hint:* Count the configurations that can be sorted after s stops.]

▸**Page 381** table entries for Straight insertion ——— 14 Jun 1977

Length: 12 $\leadsto$ 10
Space: $N \leadsto N+1$
Average: $2N^2+9N \leadsto 1.5N^2+9.5N$
Maximum: 4 $\leadsto$ 3
$N=16$: 494 $\leadsto$ 412
$N=1000$: 1985574 $\leadsto$ 1491928

Page 381 table entries for Diminishing increments ——— 15 Oct 1996

Space: $N + \epsilon \log_3 N \leadsto N + \epsilon \lg N$
Average: $15N^{1.25} \leadsto 3.9N^{7/6} + 10N \lg N + 166N$
Maximum: $cN^{1.5} \leadsto cN^{4/3}$
$N = 1000$: 137502 $\leadsto$ 128758
Footnote d: 5.2.1–8 $\leadsto$ 5.2.1–(11)
Footnote d: 5.2.1.–29 $\leadsto$ 5.2.1–29

Page 381 table entries for Merge exchange ——— 23 Jun 1996

Length: 36 $\leadsto$ 35
Average: 3.7 $\leadsto$ 2.875
Maximum: $c \leadsto 4$
$N = 16$: 1284 $\leadsto$ 939
$N = 1000$: 379104 $\leadsto$ 284366
Notes: h $\leadsto$

▸**Page 381** table entries for Median-of-3 quicksort ——— 23 Jun 1996

Average: $2.11N \leadsto 2.12N$
Footnote e: $1.57N \leadsto 1.60N$

▸**Page 381** table entry for Straight selection ——— 27 Jun 1996

Stable? Yes $\leadsto$ No

Page 381 table entries for Heapsort ——— 19 Oct 1996

Average: $0.2N \leadsto 0.01N$
Maximum: $< 26N \ln N \leadsto 24.5N \ln N$

Page 384 line 11 ——— 24 May 1995

first. $\leadsto$ first. Patent number 518240 by John K. Gore in 1894, which described another early machine for sorting cards, suggested starting with the tens column.

Page 384 line 15 ——— 15 Mar 1981

is an incidental remark which appears in an article $\leadsto$ is in a book by Robert Feindler, *Das Hollerith-Lochkarten-Verfahren* (Berlin: Reimar Hobbing, 1929), 126–130; it was also mentioned at about the same time in an article

Page 384 insert new paragraph before line −1 ——— 25 Jun 1976

In Germany, K. Zuse independently constructed a program for straight insertion sorting in 1945, as one of the simplest examples of linear list operations in his "Plankalkül" language. This pioneering work remained unpublished for nearly 30 years; see *Berichte der Gesellschaft für Math. und Datenv.* **63** (1972), part 4, 84–85.

Page 386 line −2 ——— 07 Jun 1995

92 pp. $\leadsto$ 92 pp.; *IEEE Trans.* **C-34** (1985), 296–310

▸**Page 387** line 2 ——— 25 Aug 1976

near-optional $\leadsto$ near-optimal

Page 388 line 12 ——— 24 May 1995

was being written ⇝ was first being written, compiled with the help of R. L. Rivest,

Page 391 line −3 ——— 14 May 1976

"Soundex" method, which ⇝ contemporary form of the "Soundex" method, a technique that

Page 392 top ——— 21 May 1996

line 8: omit ⇝ or adjacent except for intervening h's and w's, omit
line 12: and Łukasiewicz ⇝ Łukasiewicz, and Wachs
line 13: L222 ⇝ L222, W200
line 15: six ⇝ seven
line 16: and Lissajous ⇝ Lissajous, and Waugh

Page 392 lines −12 and −11 ——— 10 Oct 1980

Cincinnati Redlegs ⇝ Chicago White Sox

Page 394 Fig. 1 ——— 19 Apr 1977

label the downward branch coming out of box S2 with an = sign

Page 394 caption to Fig. 1 ——— 02 Mar 1977

search. ⇝ or "house-to-house" search.

Page 397 following (8) ——— 14 May 1976

formulated ⇝ popularized
inversely ... *Ecology* ⇝ approximately proportional to $1/n$. [*The Psycho-Biology of Language* (Boston, Mass.: Houghton Mifflin, 1935); *Human Behavior and the Principle of Least Effort*

Page 397 line −9 ——— 30 Sep 1996

A. D. Booth et al., ⇝ A. D. Booth, L. Brandwood, and J. P. Cleave,

Page 398 the lines before (14) ——— 25 May 1996

(Indeed, Zipf ... income.). ⇝

Page 398 lines 3–9 following (14) ——— 25 May 1996

that a more appropriate ... as $N \to \infty$. ⇝ that distribution (13) with a slightly *negative* value of θ gives a better fit to the data than Zipf's law (8). In this case the mean value

$$\overline{C}_N = H_N^{(-\theta)}/H_N^{(1-\theta)} = \frac{N^{1+\theta}}{(1+\theta)\zeta(1-\theta)} + O(N^{1+2\theta}) \qquad (15)$$

is substantially smaller than (9) as $N \to \infty$.

Distributions like (11) and (13) were first studied by Vilfredo Pareto in connection with disparities of personal income and wealth [*Cours d'Économie Politique* **2** (Lausanne: Rouge, 1897), 304–312]. If p_k is proportional to the wealth of the kth richest individual, the probability that a person's wealth exceeds or equals x times the wealth of the poorest individual is k/N when $x = p_k/p_N$. Thus, when $p_k = ck^{\theta-1}$ and $x = (k/N)^{\theta-1}$, the stated probability is $x^{-1/(1-\theta)}$; this is now called a Pareto distribution with parameter $1/(1-\theta)$.

Curiously, Pareto didn't understand his own distribution; he believed that a value of θ near 0 would correspond to a more egalitarian society than a value near 1! His error was corrected by Corrado Gini [*Atti della III Riunione della Società Italiana per il Progresso delle Scienze* (1910), reprinted in his *Memorie di metodologia statistica* **1** (Rome: 1955), 3–120], who was the first person to formulate and explain the significance of ratios like the 80-20 law (10). People still are somewhat confused by such distributions; they often speak about a "75-25 law" or a "90-10 law" as if an a-b law makes sense only when $a + b = 100$, while (12) shows that the sum $80 + 20$ is quite irrelevant.

Another discrete distribution analogous to (11) and (13) was introduced by G. Udny Yule when he studied the increase in biological species as a function of time, assuming various models of evolution [*Philos. Trans.* **B213** (1924), 21–87]. Yule's distribution applies when $\theta < 2$:

$$p_1 = c,\ p_2 = \frac{c}{2-\theta},\ p_3 = \frac{2c}{(3-\theta)(2-\theta)},\ \ldots,\ p_N = \frac{(N-1)!\,c}{(N-\theta)\ldots(2-\theta)} = \frac{c}{\binom{N-\theta}{N-1}};$$
$$c = \frac{\theta}{1-\theta}\,\frac{\binom{N-\theta}{N}}{1-\binom{N-\theta}{N}}. \qquad (16)$$

(The limiting value $c = 1/H_N$ or $c = 1/N$ is used when $\theta = 0$ or $\theta = 1$.)

Page 399 amendment to line 14 ——— 13 Feb 1995

In fact, it is always less than $\pi/2$ times the optimal value [Chung, Hajela, and Seymour, *J. Comp. Syst. Sci.* **36** (1988), 148–157]; this ratio is the best possible constant in general, since it is approached when p_j is proportional to $1/j^2$.

Page 399 lines −10, −9, −8 ——— 04 Jan 1997

(17). See ... McCabe also $\leadsto$ (17). McCabe also

Page 399 bottom line ——— 15 Sep 1996

equal. [To appear.] $\leadsto$ equal [*CACM* **19** (1976), 63–67]. However, convergence to the asymptotic limit is much slower than for the move-to-front heuristic, so move-to-front is better unless the process is prolonged [J. R. Bitner, *SICOMP* **8** (1979), 82–110]. Moreover, J. L. Bentley, C. C. McGeoch, D. D. Sleator, and R. E. Tarjan have proved that the move-to-front method never makes more than four times the total number of memory accesses made by any algorithm on linear lists, given any sequence of accesses whatever to the data, even if the algorithm knows the future; the frequency-count and transposition methods do not have this property [*CACM* **28** (1985), 202–208, 404–411].

Page 400 lines 12 and −5 ——— 28 Feb 1978

running time $\leadsto$ average running time

Page 402 caption to Table 1 ______ 27 Dec 1996

963. ↝ 963; **35** (1980), 1435–1436; J. Young and A. Potler, *Math. Comp.* **52** (1989), 221–224.

Page 403 replacement for exercise 9 ______ 25 May 1996

▸**9.** [*M27*] The text observes that the probability distributions given by (11), (13), and (16) are roughly equivalent when $0 < \theta < 1$, and that the mean number of comparisons using (13) is $\frac{\theta}{\theta+1}N + O(N^{1-\theta})$. (a) Is the mean number of comparisons equal to $\frac{\theta}{\theta+1}N + O(N^{1-\theta})$ also when the probabilities of (11) are used? (b) What about (16)?

Page 405 line 3 of exercise 19 ______ 01 Jun 1981

i, j? ↝ $i \neq j$?

Page 408 annotation on line *08* of Program B ______ 14 May 1976

rA ← ⌊rA/2⌋. ↝ rA ← ⌊rA/2⌋. (rX changes too)

▸**Page 410** lines 3 and 4 after (2) ______ 07 Nov 1984

The variance of C_N can also be expressed in terms of the variance of C'_N ↝ The variance of successful-search comparisons can also be expressed in terms of the corresponding variance for unsuccessful searches

▸**Page 410** line −7 ______ 05 Jun 1975

if and only all ↝ if and only if all

Page 413 lines −4 and −3 ______ 19 Apr 1977

if $K > K_i$ and $N > 2^k$, we ↝ if $K > K_i$ we

$\lfloor \lg (N - 2^k) \rfloor + 1$ ↝ $\lceil \lg(N - 2^k + 1) \rceil$

▸**Page 416** replacement for (9) ______ 03 Jan 1998

$$\bigl((3.6 + 0.8\phi)k + 6.2 - 5.2\phi\bigr)u \approx (7.050 \lg N + 1.08)u \quad (9)$$

for a successful search, plus $(9 - 3\phi)u \approx 4.15u$ for an unsuccessful search.

Page 417 replace lines 15–21 by two new paragraphs ______ 14 May 1976

Interpolation search is asymptotically superior to binary search. One step of binary search essentially reduces the amount of uncertainty from n to $\frac{1}{2}n$, while one step of interpolation search essentially reduces it to $\sqrt{n}$, when the keys in the table are randomly distributed. Hence interpolation search takes about $\lg\lg N$ steps, on the average, to reduce the uncertainty from N to 2. (See exercise 22.)

However, computer simulation experiments show that interpolation search does not decrease the number of comparisons enough to compensate for the extra computing time involved, unless the table is rather large. Typical files aren't sufficiently random, and the difference between $\lg\lg N$ and $\lg N$ is not substantial unless N exceeds, say, $2^{16} = 65{,}536$. Interpolation is most successful in the early stages of searching a large possibly external file; after the range has been narrowed down, binary search finishes things off more quickly. (Note that dictionary lookup by hand is essentially an external, not an internal, search. We shall discuss external searching later.)

Page 417 lines -20 to -11 ________ 14 May 1976

lines -20 and -19: is apparently ... 800 $\rightsquigarrow$ contains more than 100 pairs of values, which appear to be the beginning of a list of approximately 500

the display on lines -16 to -12 should be:

01 13 09 34 29 08 08 53 20	49 12 27
01 13 14 31 52 30	49 09 07 12
01 13 43 40 48	48 49 41 15
01 13 48 40 30	48 46 22 59 25 25 55 33 20
01 14 04 26 40	48 36

line -11: 800 $\rightsquigarrow$ 500

Page 417 lines -10 and -9 ________ 28 Dec 1995

CACM **15** (1972), 671–677, $\rightsquigarrow$ *Selected Papers on Computer Science* (Cambridge University Press, 1996), Chapter 11,

Page 418 bottom line, and line 2 of page 419 ________ 24 May 1995

It is interesting to note that Cawdrey was $\rightsquigarrow$ Cawdrey seems to have been
the last part ... order! $\rightsquigarrow$ the alphabetical order in the last part is fairly good.

Page 419 lines 13–14 ________ 02 Mar 1977

H. Bottenbruch ... He $\rightsquigarrow$ D. H. Lehmer [*Proc. Symp. Appl. Math.* **10** (1960), 180–181] was apparently the first to publish a binary search algorithm that works for all N. The next step was taken by H. Bottenbruch [*JACM* **9** (1962), 214], who

Page 419 line 22 ________ 02 Jun 1980

but ... 23). $\rightsquigarrow$ but a successful search will require about one more iteration, on the average, because of (2). Since the inner loop is performed only about $\lg N$ times, this tradeoff between an extra iteration and a faster loop does not save time unless n is extremely large. (See exercise 23.) On the other hand Bottenbruch's algorithm will find the rightmost occurrence of a given key when the table contains duplicates, and this property is occasionally important.

Page 419 lines 11 and 10 before the exercises ________ 12 Nov 1991

648], but his flowchart ... as a curiosity $\rightsquigarrow$ 648]. Binary trees similar to Fibonacci trees appeared in the pioneering work of the Norwegian mathematician Axel Thue as early as 1910 (see exercise 28). A Fibonacci tree without labels was also exhibited as a curiosity

Page 419 lines -6 through -3 ________ 24 May 1995

132]; he gave ... experiments. $\rightsquigarrow$ 132]. A correct analysis of its average behavior was not discovered until many years later (see exercise 22).

Page 422 replace exercise 22 ________ 14 May 1976

22. [*M41*] (A. C. Yao and F. F. Yao.) Show that an appropriate formulation of interpolation search requires asymptotically $\lg\lg N$ comparisons, on the average, when applied to n independent uniform random keys that have been sorted. Furthermore *all* search algorithms on such tables must make asymptotically $\lg\lg N$ comparisons, on the average.

Page 422 line 9 ——— 02 Jun 1980

necessary! ⤳ necessary on a successful search!

▸**Page 422** exercise 27 line 6 ——— 24 Jan 1979

$1 \le j \le n$ ⤳ $1 \le j \le k$

Page 422 new exercise ——— 12 Nov 1991

28. [*M23*] Define *Thue trees* T_n by means of algebraic expressions in a binary operator $*$ as follows: $T_0(x) = x * x$, $T_1(x) = x$, $T_{n+2}(x) = T_{n+1}(x) * T_n(x)$. (a) The number of leaves of T_n is the number of occurrences of x when $T_n(x)$ is written out in full. Express this number in terms of Fibonacci numbers. (b) Prove that if the binary operator $*$ satisfies the axiom

$$\bigl((x * x) * x\bigr) * \bigl((x * x) * x\bigr) = x\,,$$

then $T_m\bigl(T_n(x)\bigr) = T_{m+n-1}(x)$ for all $m \ge 0$ and $n \ge 1$.

Page 422 new exercises ——— 21 Jun 1996

▸**29.** [*22*] (Paul Feldman.) Instead of assuming that $K_1 < K_2 < \cdots < K_N$, assume only that $K_{p(1)} < K_{p(2)} < \cdots < K_{p(N)}$ where the permutation $p(1)p(2)\ldots p(N)$ is an involution, and $p(j) = j$ for all even values of j. Show that we can locate any given key K, or determine that K is not present, by making at most $2\lfloor \lg N \rfloor + 1$ comparisons.

30. [*27*] (*Involution coding.*) Using the idea of the previous exercise, find a way to arrange N distinct keys in such a way that their relative order implicitly encodes an arbitrarily given array of t-bit numbers $x_1, x_2, \ldots, x_m$, when $m \le \frac{1}{4}N + 1 - 2^t$. With your arrangement it should be possible to determine the leading k bits of x_j by making only k comparisons, for any given j, as well as to look up an arbitrary key with $\le 2\lfloor \lg N \rfloor + 1$ comparisons.

Page 424 line −8 ——— 14 May 1976

(i.e., `ROOT` $\ne \Lambda$). ⤳ (i.e., `ROOT` $\ne \Lambda$), since the necessary operations are trivial when `ROOT` $= \Lambda$.

Page 426 line 1 after the figure ——— 26 Oct 1996

$C1 - S \approx C2$ ⤳ $C1 - S$ has the same probability distribution as $C2$

▸**Page 428** line 17 ——— 14 May 1976

Algorithm L ⤳ Algorithm T

▸**Page 428** line −14 ——— 23 Feb 1987

NC_N ⤳ $NC_N - N$

Page 429 line 7 ——— 27 May 1978

to D4. ⤳ to D4. (For example, if `Q` $\equiv$ `RLINK(P)` for some `P`, we would set `RLINK(P)` $\leftarrow$ `LLINK(T)`.)

Page 431 bottom four lines ——— 22 May 1996

Empirical evidence ... theoretical ⤳ Even more startling is the empirical evidence gathered by J. L. Eppinger [*CACM* **26** (1983), 663–669, **27** (1984), 235], who found that the path length decreases slightly when a few random deletions and insertions are made, but then it *increases* until reaching a steady state after about n^2 deletion/insertion operations have been performed. This steady state is *worse* than the behavior of a random tree, when N is greater than about 150. Further study by Culberson and Munro [*Comp. J.* **32** (1989), 68–75; *Algorithmica* **5** (1990), 295–311] has led to a plausible conjecture that the average search time in the steady state is asymptotically $\sqrt{2N/9\pi}$. However, Eppinger also devised a simple modification that alternates between Algorithm D and a left-right reflection of the same algorithm; he found that this leads to an excellent steady state in which the path length is reduced to about 88% of its normal value for random trees. A theoretical

Page 432 line 7 ——— 22 May 1996

better. ⇝ better. When this idea is combined with Eppinger's symmetric deletion strategy, the steady-state path length for repeated random deletion/insertion operations decreases to about 86% of its insertion-only value.

Page 438 Fig. 16 ——— 14 Jun 1977

Insert 'a)' at the left of the top line (on the same level as the root of the top tree, the node that says `OF`); similarly, insert 'b)' at the left of the top line of the bottom tree (level with the node that says `METHOD`). Then change circles to squares in the right descendants of nodes `AN` and `AS` in the upper tree.

▸**Page 438** lines 10 and 11 from the bottom ——— 18 Sep 1997

4.75 comparisons ⇝ 4.15 comparisons
5.29 and 5.32 ⇝ 4.69 and 4.55

Page 439 lines 3–6 ——— 14 May 1976

the weights ... fairly satisfactory ⇝ the resulting maximum subtree weight, $\max\bigl(w(0,k-1), w(k,n)\bigr)$, is as small as possible. This approach can also be fairly poor, because it may choose a node with very small p_k to be the root; however, Paul J. Bayer has proved that the resulting tree will always have a weighted path length near the optimum (see exercise 36).

A more satisfactory

▸**Page 439** Fig. 17 ——— 18 Sep 1997

[The labels for minimum average cost should run from 4.0 to 4.9, not 4.6 to 5.5.]

Page 439 line −7 ——— 14 May 1976

time. ⇝ time. In fact, M. Fredman has shown that $O(n)$ units of time suffice, if suitable data structures are used [*STOC* **7** (1975), 240–244]; see K. Mehlhorn, *Data Structures and Algorithms* **1** (Springer, 1984), Section 4.2.

Page 444 should be simplified . . . . 14 May 1976

Having the external nodes in the priority queues (cf. (23)) is unnecessarily cumbersome. When I revise this material, I'll consider and perhaps switch to the similar algorithm of Garsia and Wachs cited at the end of this section.

Page 445 technical terms . . . . 11 Feb 1995

I should revise this so that the word "entropy" is explicitly mentioned, and so that the new exercise 36 is a natural generalization

Page 446 new paragraph before 'history and bibliography' ________ 13 Feb 1995

Kleitman and Saks [*SIAM J. Algeb. Discr. Methods* **2** (1981), 142–146] proved that the optimum weighted path length when $p_1 = \cdots = p_n = 0$ never exceeds the value of the optimum weighted path length that occurs when the q's have been rearranged in "sawtooth order":

$$q_0 \le q_2 \le q_4 \le \cdots \le q_{2\lfloor n/2\rfloor} \le q_{2\lceil n/2\rceil-1} \le \cdots \le q_3 \le q_1\,. \tag{26}$$

(This is the inverse of the organ-pipe order discussed in exercise 6.1–18.) In the latter case the Hu-Tucker algorithm essentially reduces to Huffman's algorithm on the weights q_0+q_1, q_2+q_3, $\ldots$; so we can improve the upper bound of Theorem G without knowing the order of the weights.

Page 447 just before the exercises ________ 15 Oct 1996

532]. $\rightsquigarrow$ 532]. A simplification was found by A. M. Garsia and M. L. Wachs, *SICOMP* **6** (1977), 622–642; see J. H. Kingston, *J. Algorithms* **9** (1988), 129–136, for a simple proof of their algorithm. See also the paper by Hu and J. D. Morgenthaler, *Lecture Notes in Comp. Sci.* **1120** (1996), 234–243, for special cases in which the optimum tree can be found in $O(n)$ steps.

Page 448 last line of exercise 6 ________ 13 Apr 1981

of C'_{n-1}? $\rightsquigarrow$ of this distribution?

Page 448 exercise 7 ________ 28 Jan 1982

rating: [*M30*] $\rightsquigarrow$ [*M25*]
line 3: element? $\rightsquigarrow$ element, given the key of that element?

Page 448 exercise 10 ________ 30 May 1995

Suppose $\rightsquigarrow$ (R. W. Floyd.) Suppose

▸**Page 449** exercise 23 ________ 15 Nov 1978

$p_1 = 5 \rightsquigarrow p_1 = 9$

Page 450 new version of exercise 30 ________ 14 May 1976

30. [*M34*] Prove that the costs of optimum binary search trees satisfy the inequality $c(i,j) - c(i,j-1) \ge c(i+1,j) - c(i+1,j-1)$ when $j \ge i+2$.

Page 450 end of exercise 33 ________ 19 Dec 1976

508–517.) $\rightsquigarrow$ 508–517. On machines that cannot make three-way comparisons at once, a program for Algorithm T will have to make two comparisons in step T2, one for equality and one for less-than; B. Sheil and V. R. Pratt have observed that these comparisons need not involve the same key, and it may well be best to have a binary tree whose internal nodes specify either an equality test *or* a less-than test but not both. This situation would be interesting to explore as an alternative to the stated problem. See also exercise 42.)

Page 450 new version of exercise 36 ———— 14 May 1976

36. [*M40*] (Paul J. Bayer, 1975.) Generalizing the upper bound of Theorem G, prove that the cost of any optimum binary search tree with nonnegative weights $(p_1, \ldots, p_n; q_0, \ldots, q_n)$ must be at most the total weight $W = w(0,n)$ times $H + 2$, where

$$H = \sum_{j=1}^{n} \frac{p_j}{W} \lg \frac{W}{p_j} + \sum_{j=0}^{n} \frac{q_j}{W} \lg \frac{W}{q_j}$$

is the entropy of the weights. In fact, the top-down procedure that repeatedly chooses roots to minimize the maximum subtree weights will yield a binary search tree satisfying this bound. Show further that the cost of the optimum binary tree search must be at least $(H - \lg(2H/e))W$.

Page 451 rating for exercise 40 ———— 13 Feb 1995

[*M50*] ⇝ [*48*]

Page 451 new exercises ———— 13 Jul 1996

▸**42.** [*25*] Modify Algorithm T so that it uses two-way branching instead of three-way branching (see exercise 6.2.1–23).

43. [*M27*] What is the average number of comparisons made by the algorithm of the previous exercise, when N keys have been inserted in random order?

Page 451 new exercise ———— 09 Sep 1996

44. [*HM42*] Analyze the average height of a random binary search tree.

Page 451 line −7 ———— 17 Sep 1996

But this ⇝ (See exercise 21.) But this

Page 451 new exercise ———— 25 May 1996

41. [*M47*] Carry out a rigorous analysis of the steady state of a binary search tree subjected to random insertions and deletions.

▸**Page 453** line −8 ———— 16 Jan 1998

$F_{27} - 1 = 196{,}417$ ⇝ $F_{28} - 1 = 317{,}810$

Page 454 replacement for bottom line ———— 22 May 1996

by exploiting the fact that the balance factor of node B in (1) was zero before the insertion.

▸**Page 456** Fig. 22 ———— 21 Sep 1976

The arrows between boxes A2 and A3 should be reversed (downward arrow on the left, upward arrow on the right). Also delete "`P` $= \Lambda$" below boxes A3 and A4, and insert the words "Leaf found" between the two arrows leading to A5.

▸**Page 456** line 2 below the figure ———— 14 May 1976

`KEY(Q)` ← `K` ⇝ `KEY(Q)` ← K

Page 457 step A10 ———— 22 May 1996

line 2: new root ⇝ new subtree root
line 3: old root ⇝ old subtree root `S`.

Page 457 line 15 ——— 28 Feb 1978

necessary. ⇝ necessary. Essentially the same method can be used if the tree is *threaded* (cf. exercise 6.2.2–2), since the balancing act never needs to make difficult changes to thread links.

▸**Page 459** line −3 ——— 16 Jan 1998

1.43684 ⇝ 1.43687

Page 460 near the bottom ——— 05 May 1983

line −6: extremely plausible ⇝ plausible
line −5: is about ⇝ should be about
lines −5 to −3: Empirical tests ... large N. ⇝ But R. W. Floyd has observed that the coefficient of $\lg N$ is unlikely to be exactly 1, because the root of the tree would then be near the median, and the roots of its two subtrees would be near the quartiles; then single and double rotation could not easily keep the root near the median. Empirical tests indicate that the true average number of comparisons needed to insert the Nth item is approximately $1.01 \lg N + 0.1$, except when N is small.

Page 461 last line before Table 1 ——— 18 Feb 1995

probable. ⇝ probable. (The probabilities that show up as .143 in Table 1 are actually equal to 1/7, for all $N \geq 7$; see exercise 11. Single and double rotations are equally likely when $N \leq 15$, but double rotations occur slightly less often when $N \geq 16$.)

Page 461 new data for Table 1 ——— 18 Feb 1995

Path length k	No rebalancing	Single rotation	Double rotation
1	.143	.000	.000
2	.152	.143	.143
3	.092	.048	.048
4	.060	.024	.024
5	.036	.010	.010
> 5	.051	.009	.008
ave 2.78	total .534	.233	.232

Page 462 empirical info in lines 9 and 10 ——— 05 Jun 1975

lines 9–10: $.144 + .153 + .144 + .144 = .585$ ⇝ $.143 + .153 + .143 + .143 = .582$
line 13: $.535 + 2(.233 + .232) = 1.5$ ⇝ $.534 + 2(.233 + .232) \approx 1.5$

Page 463 empirical info in lines 1–14 ——— 05 Jun 1975

line 1: $\lg N + 0.25$ ⇝ $1.01 \lg N + 0.1$
line 2: $11.17 \lg N + 4.8$ ⇝ $11.3 \lg N + 3.3$
line 6: $6.5 \lg N + 4.1$ ⇝ $6.6 \lg N - 3.4$
line 14: .535 ⇝ .534

▸**Page 464** line −9 ______ 04 Dec 1975

`R(P)` $\rightsquigarrow$ `RANK(P)`

▸**Page 465** line 16 ______ 04 Dec 1975

`RANK(R).` $\rightsquigarrow$ `RANK(R).` Go to C10.

▸**Page 466** line 1 ______ 16 Jan 1998

$B(P_k)$ $\rightsquigarrow$ $\mathtt{B}(P_k)$

Page 468 replacement for lines 6–12 ______ 14 May 1976

Alternatives to balanced trees. Many other ways have been proposed to organize trees so that logarithmic accessing time is guaranteed. For example, C. C. Foster [*CACM* **16** (1973), 513–517] considered the binary trees that arise when we allow the height difference of subtrees to be at most k. Such structures may be called HB(k) (meaning "height balanced"), so that ordinary balanced trees represent the special case HB(1).

Page 468 line −4 ______ 14 May 1976

(unpublished) $\rightsquigarrow$ [see Aho, Hopcroft, and Ullman, *The Design and Analysis of Computer Algorithms* (Reading, Mass.: Addison-Wesley, 1974), Chapter 4]

Page 469 just before the exercises . . . . 16 Sep 1996

Elaboration of these ideas leads to many additional flavors of balanced trees, most notably the red-black trees, also called symmetric binary B-trees or half-balanced trees [R. Bayer, Acta Informatica **1** *(1972), 290–306; L. Guibas and R. Sedgewick, FOCS* **19** *(1978), 8–21; H. J. Olivié, RAIRO Informatique Théorique* **16** *(1982), 51–71; R. E. Tarjan, Inf. Proc. Letters* **16** *(1983), 253–257]. There is also a strongly related family called hysterical B-trees or (a, b)-trees, notably $(2, 4)$-trees [D. Maier and S. C. Salveter, Inf. Proc. Letters* **12** *(1981), 199–202; S. Huddleston and K. Mehlhorn, Acta Informatica* **17** *(1982), 157–184].*

When some keys are accessed much more frequently than others, we want the important ones to be relatively close to the root, as in the optimum binary search trees of Section 6.2.2. Dynamic trees that make it possible to maintain weighted balance within a constant factor of the optimum, called biased trees, *have been developed by S. W. Bent, D. D. Sleator, and R. E. Tarjan, SICOMP* **14** *(1985), 545–568; J. Feigenbaum and R. E. Tarjan, Bell System Tech. J.* **62** *(1983), 3139–3158. The algorithms are, however, quite complicated. A much simpler self-adjusting data structure called a* splay tree, *based on ideas similar to the move-to-front and transposition heuristics discussed in Section 6.1 [B. Allen and I. Munro, JACM* **25** *(1978), 526–535; J. Bitner, SICOMP* **8** *(1979), 82–110], was developed subsequently by D. D. Sleator and R. E. Tarjan [JACM* **32** *(1985), 652–686]. Splay trees, like the other kinds of balanced trees already mentioned, support the operations of concatenation and splitting as well as insertion and deletion, and in a particularly simple way. Moreover, the time needed to access data in a splay tree is known to be at most a small constant multiple of the access time of a statically optimum tree, when amortized over any series of operations,* *and it is conjectured to be at most a constant multiple of the optimum time to access data and to perform rotations dynamically in any binary tree algorithm whatsoever.*

Randomization leads to methods that appear to be even simpler and faster than splay trees. Jean Vuillemin [CACM **23** *(1980), 229–239] introduced "Cartesian trees," in which every node has two keys (x, y). The x parts are ordered from left to right as in binary search trees; the y parts are ordered from top to bottom as in priority queue trees. C. R. Aragon and R. G. Seidel gave this data structure the more colorful name* treap, *because it neatly combines the notions of trees and heaps. Exactly one treap can be formed with n given key pairs (x_1, y_1), …, (x_n, y_n), if the x's and y's are distinct. One way to obtain it is to insert the x's by Algorithm 6.2.2T according to the order of the y's; but there is also a simple algorithm that directly inserts any new key pair into any treap. Aragon and Seidel observed [FOCS* **30** *(1989), 540–546] that if the x's are ordinary keys while the y's are chosen at random, we can be sure that the treap has the shape of a random binary search tree. In particular, a treap with random y values will be reasonably well balanced except with exponentially small probability (see exercise 5.2.2–58). Aragon and Seidel also showed that treaps can readily be biased so that, for example, a key x with relative frequency f will appear suitably near the root when it is associated with $y = U^{1/f}$, where U is a random number between 0 and 1. Treaps performed consistently better than splay trees in some experiments conducted by D. E. Knuth relating to the calculation of convex hulls [Lecture Notes in Comp. Sci.* **606** *(1992), 53–55].*

A new section 6.2.5 devoted to randomized data structures is planned for the next edition of the present book. It will discuss "skip lists" [W. Pugh, CACM **33** *(1990), 668–676] as well as treaps.*

▸**Page 470** exercise 8 ——— 16 Jan 1998

$\to 1$ ⇝ $\to -1$

Page 470 rating for exercise 9 ——— 12 Apr 1995

[*M47*] ⇝ [*HM44*]

Page 470 replacement for exercise 10 ——— 18 Feb 1995

10. [*25*] (R. C. Richards.) Show that the shape of a balanced tree can be constructed uniquely from the list of its balance factors `B(1)B(2)`...`B(`N`)` in symmetric order.

Page 470 replacement for exercise 11 ——— 29 Nov 1977

11. [*M24*] (Mark R. Brown.) Prove that when $n \ge 6$ the average number of external nodes of each of the types `+A`, `-A`, `++B`, `+-B`, `-+B`, `--B` is exactly $(n+1)/14$, in a random balanced tree of n internal nodes constructed by Algorithm A.

Page 470 exercise 16 ——— 29 Nov 1977

the root node `F` were ⇝ node `E` and the root node `F` were both

Page 471 last line of exercise 30 ——— 23 May 1996

algorithms given ⇝ algorithms given for those methods

Page 471 new exercise ______ 14 May 1976

31. [*34*] (M. L. Fredman, 1975.) Invent a representation of linear lists with the property that insertion of a new item between positions $m-1$ and m, given m, takes $O(\log m)$ units of time.

Page 471 insert quotation before Section 6.2.4 ______ 15 Mar 1981

Samuel considered the nation of Israel, tribe by tribe,
and the tribe of Benjamin was picked by lot.
Then he considered the tribe of Benjamin, family by family,
and the family of Matri was picked by lot.
Then he considered the family of Matri, man by man,
and Saul son of Kish was picked by lot.
But when they looked for Saul he could not be found.

— *1 Samuel 10:20–21*

Page 476 clarifications ______ 31 Jan 1979

line -14: new node $\rightsquigarrow$ new key
line -11: just l ... tree. $\rightsquigarrow$ just the total number of internal nodes in the tree, minus l.
line -10: nodes $\rightsquigarrow$ internal nodes
line -8: a node $\rightsquigarrow$ a node while building a tree of N keys line -7: insertion! $\rightsquigarrow$ insertion.

Page 479 new paragraphs before the exercises ______ 19 Dec 1976

Andrew Yao has proved that the average number of nodes after random insertions without the overflow feature will be $N/(m \ln 2) + O(N/m^2)$, for large N and m, so the storage utilization will be approximately $\ln 2 = 69.3$ percent [*Acta Informatica* **9** (1978), 159–170].

It is possible for many independent users to be accessing and updating different parts of a large B-tree file simultaneously without "deadlock," if the algorithms are implemented properly; see B. Samadi, *Inf. Proc. Letters* **5** (1976), 107–112.

Page 480 clarifying exercise 5 ______ 05 Jun 1975

line 3: long, $\rightsquigarrow$ long but always a multiple of 5 characters,
line 6: flowing.") $\rightsquigarrow$ flowing"; move up the key that makes the remaining two parts most nearly equal in size.)

Page 483 line -11 ______ 23 May 1996

The idea of trie memory was first published by $\rightsquigarrow$

The abstract concept of a trie to represent a family of strings was introduced by Axel Thue, in a paper about strings that do not contain adjacent repeated substrings [*Skrifter udgivne af Videnskabs-Selskabet i Christiania*, Mathematisk-Naturvidenskabelig Klasse (1912), No. 1; reprinted in Thue's *Selected Mathematical Papers* (Oslo: Universitetsforlaget, 1977), 413–477].

Trie memory for computer searching was first recommended by

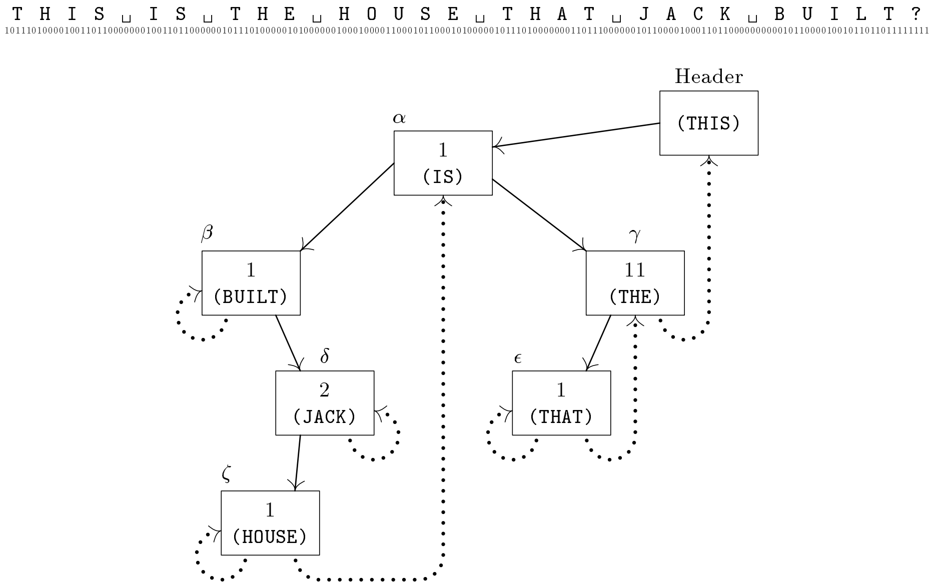
Page 486 new paragraph to follow line 9 ______________ 01 Jan 1983

T. N. Turba [*CACM* **25** (1982), 522–526] points out that it is sometimes most convenient to search for variable-length keys by having one search tree or trie for each different length.

Page 489 line −7 ______________ 23 May 1996

and so on. $\rightsquigarrow$ and so on. Alphabetic order of the keys in a digital search tree no longer corresponds to symmetric order of the nodes.

▸**Page 491** replacement for Fig. 33 ______________________ 25 Jun 1996



[Also change ‘`BUILT.`’ to ‘`BUILT?`’ at the bottom of page 490 (three places), and ‘`.`’ to ‘`?`’ in the line following Fig. 33 on page 491.]

▸**Page 492** replacement for lines 2 through 11 ______________________ 27 Dec 1979

A search in Patricia’s tree is carried out as follows: Suppose we are looking up the word `THE` (bit pattern 10111 01000 00101). We start by looking at the `SKIP` field of the root node α, which tells us to examine the first bit of the argument. It is 1, so we move to the right. The `SKIP` field in the next node, γ, tells us to look at the $1 + 11 = 12$th bit of the argument. It is 0, so we move to the left. The `SKIP` field of the next node, ϵ, tells us to look at the $(12+1)$st bit, which is 1; now we find $\texttt{RTAG} = 1$, so we go back to node γ, which refers us to the `TEXT`. The search path we have taken would occur for any argument whose bit pattern is 1xxxx xxxxx x01..., and we must check to see if it matches the unique key beginning with that pattern.

Page 500 changing the example . . . . 12 Feb 1995

I know how to improve the table in exercise 4 by packing the trie into 49 spaces, not 55, and I also know that 48 is impossible. But all the examples in this section will have to be redone for ASCII code some day, instead of the EBCDIC-like code of `MIX`*, so I’ll wait until then before refining this exercise. (See also the amendment to the answer, on page 680 below.)*

Page 502 new rating for exercise 29 ______________________ 30 May 1995

[*M46*] ↝ [*HM40*]

Page 503 line 2 — 14 May 1976

Prove that $\rightsquigarrow$ Prove that, if $n \geq 2$,

Page 504 rating for exercise 39 — 14 May 1976

[*M47*] $\rightsquigarrow$ [*M43*]

Page 507 lines 5–7 — 30 May 1995

[The birthday paradox ... 45. $\rightsquigarrow$ [The birthday paradox was discussed informally by mathematicians in the 1930s, but its origin is obscure; cf. I. J. Good, *Probability and the Weighing of Evidence* (Griffin, 1950), 38.

Page 507 line 13, add new sentence — 28 Feb 1978

Suitable techniques have been discussed by R. Sprugnoli, *CACM* **20** (1977), 841–850; **22** (1979), 104.

▸**Page 509** line 3 — 27 May 1978

$\equiv \rightsquigarrow \not\equiv$

Page 511 line −5 — 29 May 1996

(8) $\rightsquigarrow$

Page 512 line 9 — 29 May 1996

(9) $\rightsquigarrow$ (8)

Page 512 new paragraph to follow line −11 — 29 May 1996

An even better way to hash l-character or l-word keys $K = x_1 x_2 \ldots x_l$ is to compute

$$h(K) = \bigl(h_1(x_1) + h_2(x_2) + \cdots + h_l(x_l)\bigr) \bmod M, \qquad (9)$$

where each h_j is an independent hash function. This idea, introduced by J. L. Carter and M. N. Wegman in 1977, is especially efficient when each x_j is a single character, because we can then use a precomputed array for each h_j. Such arrays make multiplication unnecessary; and if M is a power of 2, we can avoid the division in (9) by substituting exclusive-or for addition. Therefore (9) certainly satisfies property (a). Moreover, Carter and Wegman proved that if the h_j are chosen at random, property (b) will hold *regardless of the input data.* (See exercise 71.)

Page 512 line −9 — 29 May 1996

division and multiplication methods $\rightsquigarrow$ methods

Page 513 replacement for line −5 — 14 May 1976

roughly a factor of M. (A precise formula is worked out in exercise 34.)

Page 518 lines 5–7 — 19 Apr 1977

using circular ... complicated. $\rightsquigarrow$ hashing `FIRE` and searching down its list, as suggested by D. E. Ferguson, since the lists are short.

▸**Page 518** corrected analysis ______ 10 Jan 1980

line 9, a new equation: $C'_N = 1 + \dfrac{N(N-1)}{2M^2} \approx 1 + \frac{1}{2}\alpha^2$

line 6 after (19): 13. $\rightsquigarrow$ 13. The method introduces a tag bit in each entry; the average number of probes needed in an unsuccessful search therefore decreases slightly, from (18) to

$$\left(1 - \frac{1}{M}\right)^N + \frac{N}{M} \approx e^{-\alpha} + \alpha. \qquad (18')$$

line 8 after (19): If separate ... $\alpha > 1$. $\rightsquigarrow$

line 11 after (19): $\frac{1}{2}$. $\rightsquigarrow$ $\frac{1}{2}$. However, it is usually preferable to use an alternative scheme that puts the first colliding elements into an auxiliary storage area, allowing lists to coalesce only when this auxiliary area has filled up; see exercise 43.

▸**Page 519** replacement for lines 10 and 11 ______ 01 Aug 1996

L2. [Compare.] If `TABLE[`i`]` is empty, go to step L4. Otherwise if `KEY[`i`]` $= K$, the algorithm terminates successfully.

▸**Page 519** the last word ______ 06 Jun 1980

$9u \rightsquigarrow 8u$

▸**Page 520** program line *13* ______ 14 May 1976

empty $\rightsquigarrow$ nonempty

▸**Page 522** line -1 ______ 14 May 1976

antially $\rightsquigarrow$ stantially

Page 524 refinements ______ 10 Jan 1980

line 1 of (30): `-M-1,1` $\rightsquigarrow$ `1-M,1`

line 1 after (30): In this $\rightsquigarrow$ Program D takes a total of $8C + 19A + B + 26 - 13S - 17S1$ units of time; modification (30) saves about $15(A - S1) \approx 7.5\alpha$ of these in a successful search. In this

Fig. 42 needs to be more accurately redrawn using the following data:

$\alpha =$	0.0	0.2	0.4	0.6	0.8	0.9	0.92	0.94	0.96	0.98	0.99
$L =$	24.0	24.9	26.3	29.3	38.0	55.5	64.3				
$D =$	23.0	25.7	28.8	32.6	38.4	43.9	45.7	47.9	51.2	56.8	62.5
$D_{\mathrm{mod}} =$	23.0	24.2	26.0	28.8	34.1	39.6	41.5	43.9	47.2	53.1	58.9

Page 526 new paragraph after line 19 ______ 29 Nov 1977

E. G. Mallach [*Comp. J.* **20** (1977), 137–140] has experimented with refinements of Brent's variation, and further results have been obtained by Gaston H. Gonnet and J. Ian Munro [*SICOMP* **8** (1979), 463–478].

Page 527 line 13 ______ 14 May 1976

similar $\rightsquigarrow$ weaker

Page 527 new paragraph to follow line 19 ———— 19 Dec 1976

Algorithm R may move some of the table entries, and this is undesirable if they are being pointed to from elsewhere. Another approach to deletions is possible by adapting some of the ideas used in garbage collection (cf. Section 2.3.5): We might keep a reference count with each key telling how many other keys collide with it; then it is possible to convert unoccupied cells to empty status when their reference drops to zero. Alternatively we might go through the entire table whenever too many deleted entries have accumulated, changing all the unoccupied positions to empty and then looking up all remaining keys, in order to see which unoccupied positions still require 'deleted' status. These procedures, which avoid relocation and work with any hash technique, were originally suggested by T. Gunji and E. Goto [*J. Information Proc.* **3** (1980), 1–12].

Page 528 lines 3–4 after (34) ———— 12 Feb 1995

purposes. ... completes the analysis of uniform hashing. $\rightsquigarrow$
purposes. In fact, double hashing is asymptotically equivalent to uniform hashing, in the limit as $M \to \infty$ (see exercise 70).

This completes our analysis of uniform hashing.

Page 529 line 20 ———— 14 May 1976

Now let $\rightsquigarrow$ By convention we also set $f(0,0) = 1$. Now let

Page 530 line 8 ———— 28 May 1996

27. $\rightsquigarrow$ 27. (For the variance, see exercises 28, 67, and 68.)

▸**Page 531** line 1 after (44) ———— 26 May 1996

0 3 2 0 1 0 0 0 2 $\rightsquigarrow$ 0 3 2 0 1 0 0 0 0 2

▸**Page 532** the line after (48) ———— 28 Feb 1978

likely we, $\rightsquigarrow$ likely, we

Page 533 middle ———— 30 May 1995

line 10: has proved $\rightsquigarrow$ proved
line 16: it seems reasonable to conjecture $\rightsquigarrow$ Ullman conjectured
append to line 18: Andrew Yao proved an asymptotic form of this conjecture by showing that the limiting cost when $N = \alpha M$ and $M \to \infty$ is $\ge \frac{1}{\alpha} \ln \frac{1}{1-\alpha}$ [*JACM* **32** (1985), 687–693].
line 19: This conjecture $\rightsquigarrow$ The strong form of Ullman's conjecture

▸**Page 534** lines 10 and 13 ———— 30 Jan 1998

$$\sum_{\pi \epsilon \Pi(A)} \rightsquigarrow \sum_{\pi \in \Pi(A)}$$

▸**Page 585** in Table 3 ———— 30 Jan 1998

line −6: 1.0216 $\rightsquigarrow$ 1.0215
line −4: 1.286 $\rightsquigarrow$ 1.268

▸**Page 586** line −4 of Table 4 ______ 30 Jan 1998

2.7 ⇝ 2.8

▸**Page 537** line −8 ______ 19 Apr 1977

access ⇝ accesses

Page 539 correction to curves S and SO in Figure 44(a) ______ 10 Jan 1980

$\alpha =$	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
$S =$	1.0	1.005	1.020	1.045	1.080	1.125	1.180	1.245	1.320	1.405	1.500
$SO =$	1.0	1.003	1.013	1.029	1.051	1.079	1.112	1.151	1.195	1.244	1.299

Page 541 line 13 from the bottom ______ 26 Jun 1996

functions. ⇝ functions. Correct analyses of Algorithm L were first published by A. G. Konheim and B. Weiss, *SIAM J. Appl. Math.* **14** (1966), 1266–1274; V. D. Podderi͡ugin, *Wissenschaftliche Zeitschrift der Technischen Universität Dresden* **17** (1968), 1087–1089.

Page 541 line −2 ______ 07 Feb 1975

until Morris's article was published in 1968 ⇝ until the late 1960s

Page 542 lines 1–6 ______ 14 May 1976

The only previous occurrence of the word among ⇝ The first published appearance of the word seems to have been in H. Hellerman's book *Digital Computer System Principles* (New York: McGraw-Hill, 1967), p. 152; the only previous occurrence among
1968! ⇝ 1967.

Page 542 new paragraph to follow line 6 ______ 01 Sep 1996

The book *Design and Analysis of Coalesced Hashing* by J. S. Vitter and W.-C. Chen (New York: Oxford Univ. Press, 1987) discusses and analyzes several instructive variants of Algorithm C.

Page 542 before the exercises . . . . 03 Sep 1996

Many advances in the theory and practice of hashing have been made since the first edition of this book, although the basic ideas discussed above still remain useful for ordinary applications. From a practical standpoint, the most important hash technique invented in the late 1970s is probably the method that Witold Lipski calls linear hashing *[Proc. 6th International Conf. on Very Large Databases (1980), 212–223], which incidentally has nothing to do with the classical technique of "linear probing." Linear hashing allows the number of hash addresses to grow and/or contract gracefully as items are inserted and/or deleted. An excellent discussion of linear hashing, including comparisons with other methods for internal searching, has been given by Per-Åke Larson in CACM* **31** *(1988), 446–457; see also W. G. Griswold and G. M. Townsend, Software Practice & Exp.* **23** *(1993), 351–367, for improvements when many large and/or small tables are present simultaneously. Linear hashing can also be used for huge databases that are distributed between many different sites on a network* *[see Litwin, Neimat, and Schneider, SIGMOD Record* **22**, *2 (1993), 327–336]. An alternative scheme called* extendible hashing, *which has the property that at most two references to external pages are needed to retrieve any record, has been proposed by R. Fagin, J. Nievergelt, N. Pippenger, and H. R. Strong [ACM Trans. on Database Systems* **4** *(1979), 315–344]. Both linear hashing and extendible hashing are preferable to the B-trees of Section 6.2.4, when the order of keys is unimportant.*

In the theoretical realm, more complicated methods have been devised by which it is possible to guarantee $O(1)$ maximum time per access, with $O(1)$ average amortized time per insertion and deletion, regardless of the keys being examined; moreover, the total storage used at any time is bounded by a constant times the number of items currently present, plus another additive constant. This result, which builds on ideas of Fredman, Komlós, and Szemerédi [JACM **31** *(1984), 538–544], is due to Dietzfelbinger, Karlin, Mehlhorn, Meyer auf der Heide, Rohnert, and Tarjan [SICOMP* **23** *(1994), 738–761].*

Page 543 exercise 10 ——— 01 Mar 1979

[*M48*] $\rightsquigarrow$ [*M38*]
Prove or disprove . . . *conjecture:* $\rightsquigarrow$
the resulting $\rightsquigarrow$ prove that the resulting

Page 544 exercise 14 (replacement for lines 3 and following) ——— 23 Feb 1979

2-bit `TAG` field and two link fields called `LINK` and `AUX`, with the following interpretation:

`TAG(P)` = 0 indicates a word in the list of available space; `LINK(P)` points to the next entry in this list, and `AUX(P)` is unused.

`TAG(P)` = 1 indicates a word in use where `P` is not the hash address of any key in the scatter table; the other fields of the word in location `P` may have any desired format.

`TAG(P)` = 2 indicates that `P` is the hash address of at least one key; `AUX(P)` points to a linked list specifying all such keys, and `LINK(P)` points to another word in the list memory. Whenever a word with `TAG(P)` = 2 is accessed during the processing of any list, we set `P` $\leftarrow$ `LINK(P)` repeatedly until reaching a word with `TAG(P)` ≤ 1. (For efficiency we might also then change prior links so that it will not be necessary to skip over the same entries again and again.)

Define suitable algorithms for inserting and retrieving keys in a combined table of this sort.

Page 544 line −10 ——— 30 May 1995

method $\rightsquigarrow$ "quadratic probing" method

Page 544 rating for exercise 23 ——— 23 Feb 1979

[*23*] $\rightsquigarrow$ [*33*]

Page 545 line −7 ——— 30 May 1995

(John Riordan.) $\rightsquigarrow$

Page 546 replacements for exercises 34(c), 35, and 36 ______ 10 Jan 1980

rating of exercise 34: [*M22*] $\leadsto$ [*M23*]
(c) Express the average number of probes for a successful search in terms of this generating function. (d) Deduce the average number of probes in an *unsuccessful* search, considering variants of the data structure in which the following conventions are used: (i) hashing is always to a list head (cf. Fig. 38); (ii) hashing is to a table position (cf. Fig. 40), but all keys except the first of a list go into a separate overflow area; (iii) hashing is to a table position and all entries appear in the hash table.

35. [*M24*] Continuing exercise 34, what is the average number of probes in an unsuccessful search when the individual lists are kept in order by their key values? Consider data structures (i), (ii), and (iii).

36. [*M23*] Continuing exercise 34(d), find the *variance* of the number of probes when the search is unsuccessful, using data structures (i) and (ii).

Page 546 new wording for exercise 37 ______ 10 Jan 1980

▸**37.** [*M29*] Eq. (19) gives the average number of probes in separate chaining when the search is successful; what is the *variance* of that number of probes?

Page 546 last line of exercise 38 ______ 12 Oct 1996

5.2.1–11. $\leadsto$ 5.2.1–(15).

Page 546 new wording for exercise 39 ______ 01 Jun 1980

39. [*M27*] Let $c_N(k)$ be the total number of lists of length k formed when Algorithm C is applied to all M^N hash sequences (35). Find a recurrence relation on the numbers $c_N(k)$ that makes it possible to determine a simple formula for the sum

$$S_N = \sum_k \binom{k}{2} c_N(k)\,.$$

How is S_N related to the number of probes in an unsuccessful search by Algorithm C?

Page 546 new wording for exercise 40 ______ 10 Jan 1980

40. [*M33*] Eq. (15) gives the average number of probes used by Algorithm C in an unsuccessful search; what is the *variance* of that number of probes?

Page 546 rating for exercise 43 ______ 08 Aug 1980

[*M42*] $\leadsto$ [*HM44*]

Page 547 rating for exercise 45 ______ 14 May 1976

[*M48*] $\leadsto$ [*M43*]

Page 548 last line of exercise 49 ______ 10 Jan 1984

$M \to \infty$. $\leadsto$ $M \to \infty$; derive formulas (57) and (58).

Page 548 exercise 56 ______ 11 Jun 1996

line 1: [*M48*] $\leadsto$ [*HM43*]
line 2: size b. $\leadsto$ size b. What is the asymptotic number of probes in a successful search when the table is full ($N = Mb$)?

Page 549 line 5 ——— 30 May 1995

analyzed in exercise 44? $\rightsquigarrow$ described by (29)? In particular, is $C_{\alpha M} \geq 1 + \frac{1}{2}\alpha + \frac{1}{2}\alpha^2 + O(\alpha^3)$ as $M \to \infty$?

Page 549 new wording for exercise 62 ——— 30 May 1995

62. [*M49*] A single hashing scheme is called *cyclic* if the increments $p_1\, p_2 \ldots p_{M-1}$ in the notation of exercise 44 are fixed for all K. (Examples of such methods are linear probing and the sequences considered in exercises 20 and 47.) An *optimum* single hashing scheme is one for which C_M is minimum, over all $(M-1)!^M$ single hashing schemes for a given M. When $M \leq 5$ the best single hashing schemes are cyclic. Is this true for all M?

Page 549 line 3 of exercise 63 ——— 30 May 1995

another? $\rightsquigarrow$ another? (This is the mean time to failure of the deletion method that simply marks cells "deleted.")

Page 549 new rating for exercise 64 ——— 30 May 1995

[*M46*] $\rightsquigarrow$ [*M41*]

Page 549 exercise 66 ——— 14 May 1976

66. $\rightsquigarrow$ ▸**66.**

Page 549 new exercises ——— 28 May 1996

67. [*M41*] If Algorithm L inserts N keys with respective hash addresses $a_1\, a_2 \ldots a_N$, let d_j be the displacement of the jth key from its home address a_j; then $C_N = 1 + (d_1 + d_2 + \cdots + d_N)/N$. Theorem P tells us that permutation of the a's has no effect on the sum $d_1 + d_2 + \cdots + d_N$. However, such permutation might drastically change the sum $d_1^2 + d_2^2 + \cdots + d_N^2$. For example, the hash sequence $1\,2 \ldots (N-1)\,(N-1)$ makes $d_1\, d_2 \ldots d_{N-1}\, d_N = 0\,0 \ldots 0\,(N-1)$ and $\sum d_j^2 = (N-1)^2$, while its reflection $(N-1)\,(N-1) \ldots 2\,1$ leads to much more civilized displacements $0\,1 \ldots 1\,1$ and $\sum d_j^2 = N-1$.

a) Which rearrangement of $a_1\, a_2 \ldots a_N$ minimizes $\sum d_j^2$?

b) Explain how to modify Algorithm L so that it maintains a least-variance set of displacements after every insertion.

c) Determine the average value of $\sum d_j^2$ with and without this modification.

68. [*M46*] What is the variance of the number of probes in a successful search by Algorithm L? In particular, what is the average of $(d_1 + d_2 + \cdots + d_N)^2$ in the notation of exercise 67?

Page 549 new exercise ——— 14 May 1976

69. [*M25*] (Andrew Yao.) Prove that all cyclic single-hashing schemes in the sense of exercise 62 satisfy the inequality $C'_{\alpha M} \geq \frac{1}{2}\bigl(1 + 1/(1-\alpha)\bigr)$. [*Hint:* Show that an unsuccessful search takes exactly k probes with probability $p_k \leq (M-N)/M$.]

Page 549 new exercise ——— 14 Feb 1995

70. [*HM43*] Prove that the expected number of probes that are needed to insert the $(\alpha M + 1)$st item with double hashing is at most the expected number needed to insert the $\bigl(\alpha M + \sqrt{O(\log M)/M}\,\bigr)$th item with uniform hashing.

Page 549 more new exercises ——————————— 30 May 1996

▸**71.** [*M28*] (*Universal hashing.*) Imagine a gigantic matrix H that has one column for every possible key K. The entries of H are numbers between 0 and $M-1$; the rows of H represent hash functions. We say that H defines a *universal family of hash functions* if any two columns agree in at most R/M rows, where R is the total number of rows. (a) Prove that if H is universal in this sense, and if we obtain a hash function h by choosing a row of H at random, than the expected size of the list containing any given key K in the method of separate chaining (Fig. 38) will be $\le 1+N/M$, after we have inserted any set of N distinct keys K_1, K_2, $\ldots$, K_N. (b) Suppose each h_j in (9) is a randomly chosen mapping from the set of all characters to the set $\{0,1,\ldots,M-1\}$. Show that this corresponds to a universal family of hash functions. (c) Would the result of (b) still be true if $h_j(0)=0$ for all j, but $h_j(x)$ is random for $x\ne 0$?

72. [*M26*] (Carter and Wegman.) Show that part (b) of the previous exercise holds even when the h_j are not completely random functions, but they have either of the following special forms: (i) Let x_j be the binary number $(b_{j(n-1)}\ldots b_{j1}b_{j0})_2$. Then $h_j(x_j)=(a_{j(n-1)}b_{j(n-1)}+\cdots+a_{j1}b_{j1}+a_{j0}b_{j0}) \bmod M$, where each a_{jk} is chosen randomly modulo M. (ii) Let M be prime and assume that $0\le x_j<M$. Then $h_j(x_j)=(a_jx_j+b_j) \bmod M$, where a_j and b_j are chosen randomly modulo M.

73. [*M29*] Let H define a universal family of hash functions. Prove or disprove: Given any N distinct columns, and any row chosen at random, the expected number of zeros in those columns is $O(1)+O(N/M)$. [Thus, every list in the method of separate chaining will have this expected size.]

74. [*M26*] Prove or disprove the following statements about the hash function h of (9), when the h_j are independent random functions: (a) The probability that $h(K)=m$ is $1/M$, for all $0\le m<M$. (b) If $K\ne K'$, the probability that $h(K)=m$ and $h(K')=m'$ is $1/M^2$, for all $0\le m,m'<M$. (c) If K, K', and K'' are distinct, the probability that $h(K)=m$, $h(K')=m'$, and $h(K'')=m''$ is $1/M^3$, for all $0\le m,m',m''<M$. (d) If K, K', K'', and K''' are distinct, the probability that $h(K)=m$, $h(K')=m'$, $h(K'')=m''$, and $h(K''')=m'''$ is $1/M^4$, for all $0\le m,m',m'',m'''<M$.

▸**75.** [*M21*] Suggest a way to modify (9) for keys with variable length, preserving the properties of universal hashing.

Page 549 another new exercise ——————————— 11 Jun 1996

76. [*M22*] Let H define a universal family of hash functions from 32-bit keys to 16-bit keys. (Thus H has 2^{32} columns, and $M=2^{16}$, in the notation of exercise 71.) A 256-bit key can be regarded as the concatenation of eight 32-bit parts $x_1x_2x_3x_4x_5x_6x_7x_8$; we can map it into a 16-bit address with the hash function

$$h_4\bigl(h_3\bigl(h_2(h_1(x_1)h_1(x_2))h_2(h_1(x_3)h_1(x_4))\bigr)h_3\bigl(h_2(h_1(x_5)h_1(x_6))h_2(h_1(x_7)h_1(x_8))\bigr)\bigr),$$

where h_1, h_2, h_3, and h_4 are randomly and independently chosen rows of H. (Here, for example, $h_1(x_1)h_1(x_2)$ stands for the 32-bit number obtained by concatenating $h_1(x_1)$ with $h_1(x_2)$.) Prove that the probability is less than 2^{-14} that two distinct keys hash to the same address. [This scheme requires substantially fewer random choices than (9).]

Page 549 another quotation, to be put above the other ——————— 16 Jan 1977

She made a hash of the proper names, to be sure.
— GRANT ALLEN (*The Tents of Shem*, 1889)

Page 551 lines 2–10 ——— 31 May 1995

machine! Computers ... People climb $\leadsto$ machine! People climb

Page 555 lines 8 and 13 ——— 14 May 1976

[*Acta Informatica* **3** (to appear).] Bentley has recently improved on $\leadsto$ [*Acta Informatica* **4** (1974), 1–9.] Then Bentley modified
Section 6.2.2. $\leadsto$ Section 6.2.2. [*CACM* **18** (1975), 509–517.]

Page 555 future revisions .... 14 May 1976

The second edition will delete the material on post-office trees and pay more attention to k-D trees. In the binary tree line, I also plan to discuss Burkhard's search procedure and its analysis by Dubost and Trousse [cf. Stanford CS report of September 1975]—this is related to the generalized tries discussed briefly on page 565.

▸**Page 558** Fig. 46 ——— 29 Mar 1984

The three rightmost notches should move left one position (to start with 'SALT')

▸**Page 560** the superimposed codes for Toffee Bars ——— 02 Feb 1998

0010111101 $\leadsto$ 0010101101

▸**Page 561** the numbers in (5) should be respectively: ——— 01 Jul 1975

.07948358; .00708659; .00067094; .00006786; .00000728; .00000082.

Page 561 new paragraph to insert after line 18 ——— 02 Mar 1977

If carefully selected nonrandom codes are used, it is possible to avoid false drops in superimposed coding, as shown by W. H. Kautz and R. C. Singleton, *IEEE Trans.* **IT-10** (1964), 363–377; one of their constructions appears in exercise 16.

Page 563 line 9 ——— 25 Aug 1976

his Ph.D. thesis (Stanford University, 1973). $\leadsto$ *SICOMP* **5** (1976), 19–50.

▸**Page 563** quick, Watson, near the top ——— 27 May 1978

line 11: the `N**D*E` $\leadsto$ the form `N**D*E`
line 12: {`NEEDLE`, `NODDLE`, `NOODLE`} $\leadsto$
{`NEEDLE`, `NIDDLE`, `NODDLE`, `NOODLE`, `NUDDLE`}

Page 564 replacement for (7) ——— 03 Jan 1997

$$\begin{array}{ll} * \ 0 \ 0 \ 1 \to 1 & * \ 1 \ 1 \ 0 \to 5 \\ 0 \ * \ 0 \ 0 \to 2 & 1 \ * \ 1 \ 1 \to 6 \\ 1 \ 0 \ * \ 0 \to 3 & 0 \ 1 \ * \ 1 \to 7 \\ 1 \ 1 \ 0 \ * \to 4 & 0 \ 0 \ 1 \ * \to 8 \end{array} \tag{7}$$

Page 564 line 2 after (7) ——— 03 Jan 1997

locations 1, 2, 3, 5, and 7 $\leadsto$ locations 1, 2, 5, 7, and 8

Page 564 lines 15 thru −13 ______ 17 Feb 1997

Some further ... ≤ 2. $\rightsquigarrow$ In 1997, A. E. Brouwer found an attractive way to compress 8 bits to 5, with a mapping analogous to (7). Every 8-bit byte belongs to exactly one of the 32 classes

$$\begin{array}{llll}
0{*}000{*}0{*} & 01{*}0{*}{*}11 & 00{*}11{*}{*}1 & {*}11{*}{*}101\\
1{*}000{*}0{*} & 11{*}0{*}{*}11 & 10{*}11{*}{*}1 & {*}11{*}{*}010\\
0{*}010{*}0{*} & 01{*}1{*}{*}11 & 00{*}0{*}01{*} & {*}10{*}0{*}10\\
1{*}010{*}0{*} & 11{*}1{*}{*}11 & 10{*}0{*}01{*} & {*}10{*}1{*}01\\
0{*}10{*}1{*}0 & 0{*}1{*}000{*} & {*}01{*}01{*}1 & {*}0{*}1001{*}\\
1{*}10{*}1{*}0 & 1{*}1{*}000{*} & {*}10{*}10{*}0 & {*}0{*}0100{*}\\
0{*}11{*}1{*}0 & 0{*}0{*}11{*}0 & {*}00{*}011{*} & {*}0{*}011{*}1\\
1{*}11{*}1{*}0 & 1{*}0{*}11{*}0 & {*}11{*}100{*} & {*}0{*}110{*}0
\end{array}. \qquad (*)$$

The $*$'s in this design are arranged in such a way that there are 3 in each row and 12 in each column. Exercise 17 explains how to obtain similar schemes that will compress records having, say, $m = 4^r$ bits into addresses having $n = 3^r$ bits.

▸**Page 565** line 13 ______ 17 Feb 1997

specified $\rightsquigarrow$ unspecified

▸**Page 566** replacement for Eq. (11) ______ 02 Mar 1977

$$\{0,4,16,21,22,24\},\ \{1,5,17,22,23,25\},\ \ldots,\ \{30,3,15,20,21,23\} \qquad (11)$$

two lines later: $(1,5,17,22,23,25) \rightsquigarrow (0,4,16,21,22,24)$

Page 567 following line 3 . . . . 01 Sep 1996

Among the significant data structures I plan to include in the next edition of this book are E. M. McCreight's "priority search trees," [SICOMP **14** *(1985), 257–276], which are especially designed to represent intersections of dynamically changing families of intervals.*

▸**Page 567** line 8 ______ 22 May 1997

68 (1963) $\rightsquigarrow$ **82** (1963)

Page 569 replacement for lines 1–7 ______ 17 Feb 1997

8. [*M32*] (R. L. Rivest.) Consider the set $Q_{t,m}$ of all $2^t\binom{m}{t}$ basic m-bit queries like (4) in which there are exactly t specified bits. Given a set S of m-bit records, let $f_t(S)$ denote the number of queries in $Q_{t,m}$ whose answer contains a member of S; and let $f_t(s,m)$ be the minimum $f_t(S)$ over all such sets S having s elements, for $0 \le s \le 2^m$. By convention, $f_t(0,0) = 0$ and $f_t(1,0) = \delta_{t0}$.

a) Prove that, for all $t \ge 1$ and $m \ge 1$, and for $0 \le s \le 2^m$,

$$f_t(s,m) = f_t(\lceil s/2\rceil, m-1) + f_{t-1}(\lceil s/2\rceil, m-1) + f_{t-1}(\lfloor s/2\rfloor, m-1).$$

▸**Page 569** line 12 ______ 17 Feb 1997

$f_t(2^r,m) \rightsquigarrow f_t(2^{m-r},m)$

▸**Page 570** line 6 ______ 02 Mar 1977

systems or $\rightsquigarrow$ systems on

▸**Page 570** line 5 of exercise 11 ——— 27 Feb 1997

is exactly $\leadsto$ in exactly

Page 570 new exercise ——— 02 Mar 1977

16. [*25*] (W. H. Kautz and R. C. Singleton.) Show that a Steiner triple system of order v can be used to construct $v(v-1)/6$ codewords of v bits each such that no codeword is contained in the superposition of any two others.

Page 570 new exercise ——— 03 Jan 1997

▸**17.** [*M35*] (*Associative block designs.*) A set of m-tuples like (7), with exactly $m-n$ $*$'s in each of 2^n rows, is called an ABD(m,n) if every column contains the same number of $*$'s and if every pair of rows has a "mismatch" (0 versus 1) in some column. Every m-bit binary number will then match exactly one row. For example, (7) is an ABD$(4,3)$.

a) Prove that an ABD(m,n) is impossible unless m is a divisor of $2^{n-1}n$ and $n^2 \geq 2m(1-2^{-n})$.

b) A row of an ABD is said to have *odd parity* if it contains an odd number of 1s. Show that, for every choice of $m-n$ columns in an ABD(m,n), the number of odd-parity rows with $*$'s in these columns equals the number of even-parity rows. In particular, each pattern of asterisks must occur in an even number of rows.

c) Find an ABD$(4,3)$ that cannot be obtained from (7) by permuting and/or complementing columns.

d) Construct an ABD$(16,9)$.

e) Construct an ABD$(16,10)$. Start with the ABD$(16,9)$ of part (d), instead of the ABD$(8,5)$ of $(*)$.

Page 570 new exercises ——— 17 Feb 1997

18. [*M22*] Analyze the ABD$(8,5)$ of $(*)$, as (7) has been analyzed in (8): How many of the 32 locations must be searched for an average query with k bits unspecified? How many must be searched in the worst case?

19. [*M47*] Find all ABD(m,n) when $n=5$ or $n=6$.

Page 570 in the future . . . . 24 Mar 1995

I'm contemplating a new Section 6.6 on Persistent Data Structures, which represent changing information in such a way that the past history can be reconstructed efficiently. Relevant early references include J. K. Mullin, Comp. J. **24** *(1981), 367–373; M. H. Overmars, Lecture Notes in Comp. Sci.* **156** *(1983), Chapter 9; E. W. Myers, ACM Symp. Principles of Prog. Lang.* **11** *(1984), 66–75; B. Chazelle, Inf. and Control* **63** *(1985), 77–99; D. Dobkin and J. I. Munro, J. Algorithms* **6** *(1985), 455–465; R. Cole, J. Algorithms* **7** *(1986), 202–220; D. Field, Inf. Proc. Letters* **24** *(1987), 95–96; C. W. Fraser and E. W. Myers, ACM Trans. Prog. Lang. and Systems* **9** *(1987), 277–295; J. R. Driscoll, N. Sarnak, D. D. Sleator, and R. E. Tarjan, J. Comp. Syst. Sci.* **38** *(1989), 86–124; R. B. Dannenberg, Software Practice & Exp.* **20** *(1990), 109–132; J. R. Driscoll, D. D. K. Sleator, and R. E. Tarjan, JACM* **41** *(1994), 943–959.*

Page 571 replacement for lines 9–12 ——— 25 Mar 1995

3. See W. J. LeVeque, *Topics in Number Theory* **2** (Reading, Mass.: Addison-Wesley, 1956), Chapter 3; P. Ribenboim, *13 Lectures on Fermat's Last Theorem* (Springer, 1979); Andrew Wiles, *Annals of Math.* **141** (1995), 443–551.

Page 575 the coding of answer 16 ——— 04 Jan 1997

line 1: key, using `MIX` ... uses $\leadsto$ key uses
lines 3 and following: change `␣␣`, `␣.`, `␣,`, `␣(`, `␣)`, `␣+`, `␣-`, `␣*`, `␣/` respectively to `␣0`, `␣1`, `␣2`, `␣3`, `␣4`, `␣5`, `␣6`, `␣7`, `␣8`.

Page 576 lines 1–3 ——— 05 May 1981

So we may ... $(p-1)/2$. $\leadsto$ In general if f is any divisor of $p-1$ and d is any divisor of $\gcd(f,n)$, we can similarly determine $(n/d) \bmod f$ by looking up the value of $b^{(p-1)/f}$ in a table of length f/d. If $p-1$ has the prime factors $q_1 \le q_2 \le \cdots \le q_t$ and if q_t is small, we can therefore compute n rapidly by finding the digits from right to left in its mixed-radix representation, for radices $q_1, \ldots, q_t$. (This idea is due to R. L. Silver, 1975; see S. C. Pohlig and M. Hellman, *IEEE Transactions* **IT-24** (1978), 106–110.)

John M. Pollard discovered an elegant way to compute discrete logs with about $O(\sqrt{p})$ operations mod p, requiring very little memory, based on the theory of random mappings. See *Math. Comp.* **32** (1978), 918–924, where he also suggests another method based on numbers $n_j = r^j \bmod p$ that have only small prime factors.

Asymptotically faster methods are discussed in exercise 4.5.4–39.

▸**Page 576** lines 7 and 8 of answer 19 ——— 27 Sep 1997

exercises 16 and 17 $\leadsto$ exercises 17 and 18

Page 576 new paragraph after answer 19 ——— 12 Nov 1976

A similar algorithm can be used to find $\max\{x_i + x_j \mid x_i + x_j \le c\}$; or to find, e.g., $\min\{x_i + y_j \mid x_i + y_j > t\}$ given t and two sorted files $x_1 \le \cdots \le x_m$, $y_1 \le \cdots \le y_n$.

Page 576 line -6 ——— 19 Dec 1976

induction. $\leadsto$ induction; `STELA`, an alternative spelling of 'stele'; `LATES`, a Nile perch.

Page 577 answer 22 ——— 31 May 1995

line -3: trees $\leadsto$ free trees
lines -2 and -1: H. I. Scoins, ... 60 $\leadsto$ J. Hopcroft and R. E. Tarjan, in *Complexity of Computer Computations* (New York: Plenum, 1972), 140–142

▸**Page 578** lines 9 and 10 ——— 26 Sep 1997

$(\log n) \leadsto (\log N)$ (twice)

▸**Page 579** answer 7, line 3 ——— 27 May 1978

$> B_k$ and append $(B_k + 1) \leadsto \ge k - B_k$ and append $k - B_k$

Page 579 last line of answer 7 ——— 25 May 1995

exercise. $\leadsto$ exercise. [The c inversion table was discussed by Rodriguez, *J. de Math.* **4** (1839), 236–240; the C inversion table appears in Netto's *Lehrbuch der Combinatorik* (1901), §5.]

Page 579 line -3 ——— 24 Jun 1996

10.) $\leadsto$ 10. Further properties were found many years later by Guilbaud and Rosenstiehl, who called Fig. 1 the "permutahedron"; see exercise 12.)

▸**Page 580** line 1 ——— 06 Jan 1997

190 $\rightsquigarrow$ **190**

Page 580 line 1 of answer 12 ——— 24 Jun 1996

[C. Berge, ... 117.] $\rightsquigarrow$ [G. Th. Guilbaud and P. Rosenstiehl, *Math. et Sciences Humaines* **4** (1963), 9–33.]

Page 580 answer 13 ——— 14 May 1976

b_{m-1} $\rightsquigarrow$ b_{m-1}, b_{m+1}

Page 580 at the end of answer 14 ——— 31 May 1995

$n \ge 1.]$ $\rightsquigarrow$ $n \ge 1$. Finite versions of Euler's infinite sum are discussed by Knuth and Paterson in *Fibonacci Quarterly* **16** (1978), 198–212.]

Page 580 addendum to answer 16 ——— 26 Mar 1995

Notes: The second identity is the limit as $n \to \infty$ of the q-nomial theorem, exercise 1.2.6–58. The first identity, similarly, is the limit as $r \to \infty$ of the dual form of that theorem, proved in the answer to that exercise. Let $n!_q = \prod_{k=1}^{n}(1 + q + \cdots + q^{k-1})$, and let $\exp_q(z) = \sum_{n=0}^{\infty} z^n/n!_q$. The first identity tells us that $\exp_q(z)$ is equal to $1/\prod_{k=0}^{\infty}(1 - q^k z(1-q))$ when $|q| < 1$; the second tells us that it equals $\prod_{k=0}^{\infty}(1 + q^{-k}z(1-q^{-1}))$ when $|q| > 1$. The resulting formal power series identity $\exp_q(z)\exp_{q^{-1}}(-z) = 1$ implies that

$$\sum_{k=0}^{n} \frac{(-1)^k q^{k(k-1)/2}}{(1-q)\ldots(1-q^k)\,(1-q)\ldots(1-q^{n-k})} = \delta_{n0}, \qquad \text{integer } n \ge 0.$$

Page 580 replacement for answer 17 ——— 14 May 1995

17.	0 0 0 0	0 1 0 0	0 0 1 0	0 0 0 1
	1 1 0 1	1 2 0 1	1 0 2 1	1 0 1 2
	1 0 1 0	0 1 1 0	0 1 2 0	0 1 0 2
	1 0 1 1	0 1 1 1	0 1 2 1	0 1 1 2
	1 0 0 1	0 1 0 1	0 0 1 1	0 0 1 2
	2 0 1 2	0 2 1 2	0 1 2 2	0 1 2 3

Page 581 lines 15–16 ——— 14 May 1976

Zolnowsky (to appear) $\rightsquigarrow$ Zolnowski, *Discrete Math.* **9** (1974), 293–298.

Page 581 replacement for answer 19 ——— 14 May 1976

19. The number of inversions is $\sum_{0<i<j<n}(\lfloor mj/n\rfloor - \lfloor mi/n\rfloor - \lfloor m(j-i)/n\rfloor) = \sum_{0<i<j<n}[mj \bmod n < mi \bmod n] = \sum_{0<r<n}\lfloor mr/n\rfloor(r - (n-r) - (n-r-1))$, which can be transformed to $\frac{1}{4}(n-1)(n-2) - \frac{1}{4}n\sigma(m,n,0)$. [*Crelle* **198** (1957), 162–166.]

Page 581 additional paragraph for answer 20 ——— 28 Apr 1996

Jacobi's identity can be proved rapidly as follows. Since

$$\prod_{k=1}^{n}(1 - u^k v^{k-1}) = (-1)^n u^{\binom{n+1}{2}} v^{\binom{n}{2}} \prod_{k=1}^{n}(1 - u^{-k}v^{1-k}),$$

the q-nomial theorem of exercise 1.2.6–58 with $q = uv$ tells us that

$$\begin{aligned}\prod_{k=1}^{n}(1-u^kv^{k-1})(1-u^{k-1}v^k) &= (-1)^n u^{\binom{n+1}{2}} v^{\binom{n}{2}} \prod_{k=-n+1}^{n}(1-u^{k-1}v^k)\\ &= (-1)^n u^{\binom{n+1}{2}} v^{\binom{n}{2}} \sum_j \binom{2n}{j}_{uv} (uv)^{\binom{k}{2}} (-u^{-n}v^{1-n})^k\\ &= \sum_j \binom{2n}{n+j}_{uv} (-1)^j u^{\binom{j}{2}} v^{\binom{j+1}{2}}.\end{aligned}$$

Multiply both sides by $\prod_{k=1}^{n}(1-u^kv^k) = \prod_{k=1}^{n}(1-q^k)$ and note that, for fixed j, we have $\binom{2n}{n+j}_q \prod_{k=1}^{n}(1-q^k) = 1 + O(q^{n+1-|j|})$. Jacobi's identity follows as $n \to \infty$.

Page 581 new answer ——— 17 Aug 1995

22. (a) Arrange the numbers $\{1, 2, \ldots, n\}$ in a circle as on the face of a clock, and point at 1. Then for $j = n$, $n-1$, ..., 1 (in this order), move the pointer counterclockwise $h_j + 1$ steps, remove the number pointed to from the circle, and call it a_j.

(b) Each i is counted as often as the sequence $a_i\,a_{i+1} \ldots a_n$ wraps around; this is the number of times that $a_j > a_{j+1}$ for $j \geq i$. Therefore each j with $a_j > a_{j+1}$ corresponds to the indices 1, ..., j being counted once. [Guo-Niu Han, *Advances in Math.* **105** (1994), 28–29; an equivalent result had been obtained by Rawlings, see the next exercise.]

Page 581 new answer ——— 27 Apr 1996

23. Suppose, for example, that $n = 5$ and $a_1\,a_2\,a_3\,a_4\,a_5 = 3\,1\,4\,2\,5$. The number of missed shots before each death must then be $2 + 5k_1$, $2 + 4k_2$, $1 + 3k_3$, $1 + 2k_2$, k_5, for some nonnegative integers k_j. Note that the dual permutation 1 4 2 5 3 has h-table 0 1 1 2 2 in the notation of the previous exercise. In general, the probability of obtaining $a_1\,a_2\,\ldots\,a_n$ will be

$$\sum_{k_1,\ldots,k_n \geq 0} q_1^{h_n+nk_1} p_1 q_2^{h_{n-1}+(n-1)k_2} p_2 \ldots q_n^{h_1+k_1} p_n$$
$$= \frac{1-q_1}{1-q_1^n}\,\frac{1-q_2}{1-q_2^{n-1}} \cdots \frac{1-q_n}{1-q_n^1} q_1^{h_n} q_2^{h_{n-1}} \ldots q_n^{h_1},$$

where $p_j = 1 - q_j$ is the probability of fatality after $j-1$ deaths, and $h_1\,h_2\,\ldots\,h_n$ corresponds to the dual of $a_1\,a_2\,\ldots\,a_n$. In particular, when $p_1 = \cdots = p_n = p = 1-q$, the probability is $q^{h_1+\cdots+h_n}/G_n(q)$. The least likely order is therefore $n \ldots 2\,1$. [J. Treadway and D. Rawlings, *Math. Mag.* **67** (1994), 345–354; Rawlings generalized the process to multiset permutations in *Int. J. Math. & Math. Sci.* **15** (1992), 291–312.]

Page 581 new answers ——— 13 May 1995

24. Let $a_0 = 0$, and say that a *generalized descent* occurs at $j < n$ if $a_j > t(a_{j+1})$. Inserting n between a_{j-1} and a_j causes a new generalized descent iff $a_{j-1} \leq t(a_j) < n$. Suppose this occurs when j has the values $j_1 > j_2 > \cdots > j_k > 0$; let the other values of j be $j_n > j_{n-1} > \cdots > j_{k+1}$. Then $j_n = n$, and it can be shown that the generalized index increases by $n-k$ when n is inserted just before a_{j_k}. [The special case in which $t(j) = j + d$ for some $d \geq 0$ is due to D. Rawlings, *J. Combinatorial Theory* **A31** (1981), 175–183; he generalized this special case to multiset permutations in *Linear and Multilinear Algebra* **10** (1981), 253–260.]

This exercise defines $n!$ different statistics on permutations, each of which has generating function $G_n(z)$. We can define many more such statistics by generalizing Russian roulette as follows: After $j-1$ deaths, the person who begins the next round of shooting is $f_j(a_1,\ldots,a_{j-1})$, where f_j is an arbitrary function taking values in $\{1,\ldots,n\}\setminus\{a_1,\ldots,a_{j-1}\}$. [Cf. Guo-Niu Han, *Calcul Denertien* (thesis, Univ. Strasbourg, 1992), Part 1.3, §7.]

25. (a) If $a_1 < a_n$, $h(\alpha)$ has exactly as many inversions as α, because the elements of α_j now invert x_j instead of a_n. But if $a_1 > a_n$, $h(\alpha)$ has $n-1$ fewer inversions, because x_j loses its inversion of a_n and of each element in α_j. Therefore if we set $b_n = a_n$ and recursively let $b_1\ldots b_{n-1} = f\bigl(h(\alpha)\bigr)$, the permutation $f(\alpha) = b_1\ldots b_n$ has the desired properties. We have $f(1\,9\,8\,2\,6\,3\,7\,4\,5) = 9\,1\,2\,6\,3\,8\,7\,4\,5$, and $f^{-1}(1\,9\,8\,2\,6\,3\,7\,4\,5) = 1\,9\,2\,6\,8\,7\,3\,4\,5$.

(b) The key point is that $\operatorname{inv}(\alpha) = \operatorname{inv}(\alpha^{-1})$ and $\operatorname{ind}(\alpha^{-1}) = \operatorname{ind}\bigl(f(\alpha)^{-1}\bigr)$. Therefore if $\alpha_1 = \alpha^{-1}$, $\alpha_2 = f(\alpha_1)$, $\alpha_3 = \alpha_2^{-1}$, $\alpha_4 = f^{-1}(\alpha_3)$, and $\alpha_5 = \alpha_4^{-1}$, we have

$$\begin{aligned}\operatorname{inv}(\alpha_5) &= \operatorname{inv}(\alpha_4) = \operatorname{ind}(\alpha_3) = \operatorname{ind}(\alpha_2^{-1}) = \operatorname{ind}(\alpha_1^{-1}) = \operatorname{ind}(\alpha);\\ \operatorname{ind}(\alpha_5) &= \operatorname{ind}(\alpha_4^{-1}) = \operatorname{ind}(\alpha_3^{-1}) = \operatorname{ind}(\alpha_2) = \operatorname{inv}(\alpha_1) = \operatorname{inv}(\alpha).\end{aligned}$$

[*Math. Nachrichten* **83** (1978), 143–159. See Garsia and Gessel, *Advances in Math.* **31** (1979), 288–305, for related results.]

Page 581 new answer ——— 06 Jun 1995

26. (Solution by Doron Zeilberger.) The average of inversions times index is

$$\frac{1}{n!}\sum_{\text{permutations}}\ \sum_{1\le j<k\le n}\ \sum_{1\le l<n}[a_j > a_k]\,l\,[a_l > a_{l+1}],$$

which is a polynomial in n of degree ≤ 4. Evaluating this sum for $1\le n\le 5$ gives the respective values 0, $\frac{1}{2}$, $\frac{6}{2}$, $\frac{21}{2}$, $\frac{55}{2}$; so the polynomial must be $\frac{1}{8}n(n-1)+\frac{1}{16}n^2(n-1)^2$. Subtracting $\operatorname{mean}(g_n)^2$ and dividing by $\operatorname{var}(g_n)$ gives the answer $9/(2n+5)$ for $n\ge 2$, by (12) and (13).

Page 581 new answers ——— 28 Apr 1996

27. Interchanging two adjacent elements changes the total displacement by 0 or ± 2; hence $\operatorname{td}(a_1\,a_2\,\ldots\,a_n) \le 2\operatorname{inv}(a_1\,a_2\,\ldots\,a_n)$.

We can also prove that $\operatorname{td}(a_1\,a_2\,\ldots\,a_n) \ge \operatorname{inv}(a_1\,a_2\,\ldots\,a_n)$. Suppose j is the smallest element out of place, and let $a_k = j$. Let l be maximum with $l<k$ and $a_l \ge k$. Interchanging a_l with a_k reduces the inversions by $2(k-l)-1$, and reduces the total displacement by $2(k-l)$. Therefore if m repetitions of this algorithm are needed to sort a given permutation $a_1\,a_2\,\ldots\,a_n$, we have $\operatorname{td}(a_1\,a_2\,\ldots\,a_n) = \operatorname{inv}(a_1\,a_2\,\ldots\,a_n)+m$.

Note: The average total displacement of a random permutation is $(n^2-1)/3$. The generating function for total displacement does not appear to have a simple form.

28. We can obtain π as a product of $\operatorname{inv}(\pi)$ transpositions τ_j, where τ_j interchanges j and $j+1$. For example, the path $1234 \to 1324 \to 1342 \to 3142$ in Fig. 1 corresponds to τ_2, then τ_3, then τ_1; hence $3\,1\,4\,2 = \tau_1\tau_3\tau_2$. Therefore $\pi\pi'$ is obtainable from π' by making $\operatorname{inv}(\pi)$ transpositions, each of which changes the number of inversions by ± 1. It follows that $\operatorname{inv}(\pi\pi') \le \operatorname{inv}(\pi)+\operatorname{inv}(\pi')$. If equality holds, each transposition adds a new inversion, hence $E(\pi\pi') \supseteq E(\pi')$.

Conversely, if $E(\pi\pi') \supseteq E(\pi')$, we want to show that some sequence of $|E(\pi\pi')| - |E(\pi')| = \operatorname{inv}(\pi\pi')-\operatorname{inv}(\pi')$ transpositions will transform π' to $\pi\pi'$. Such transpositions define π, so this will prove that $\mathrm{inv}(\pi) \le \mathrm{inv}(\pi\pi') - \mathrm{inv}(\pi')$; hence equality must hold. Suppose, for example, that $\pi' = 3\,1\,4\,5\,9\,2\,6\,8\,7$ and that $E(\pi\pi') \supseteq E(\pi')$. If $E(\pi\pi')$ does not contain $(4,1)$ or $(5,4)$ or $(9,5)$ or $(6,2)$ or $(8,6)$, then $\pi\pi'$ must be equal to π'. Otherwise $E(\pi\pi')$ contains one of them, say $(9,5)$; then $E(\pi\pi')$ contains $E(\tau_4\pi') = E(3\,1\,4\,9\,5\,2\,6\,8\,7)$. In this way we can prove the result by induction on $|E(\pi\pi')| - |E(\pi')|$.

Page 583 last three lines of answer 16 ——— 15 May 1995

Sylvester ... result. $\rightsquigarrow$ F. Franklin [*Amer. J. Math.* **5** (1882), 268–269]. (See also Pólya and Alexanderson, *Elemente der Mathematik* **26** (1971), 102–109.) Multiplying the generating functions for $j = 1, 2, \ldots$ gives the desired formula for inversions of multiset permutations, which MacMahon published in *Proc. London Math. Soc.* (2) **15** (1916), 314–321.

▸**Page 583** answer 18 ——— 15 May 1995

line -5 of page 583: 5.1.1–17 $\rightsquigarrow$ 5.1.1–25
line -4 of page 583: given in Section 5.2.1 $\rightsquigarrow$ following 5.1.1–(14)
line 2 of page 584: inserting the elements of q_j in nonincreasing order $\rightsquigarrow$ giving all elements of q_j the subscript j
line 3 of page 584: $J(a_1 \ldots a_n) \rightsquigarrow \mathrm{ind}(a_1 \ldots a_n)$

▸**Page 584** second displayed formula of answer 20 ——— 01 Aug 1997

$(-1)^n \rightsquigarrow$

Page 584 new paragraph to follow answer 20 ——— 31 May 1995

A succinct rendition of this combinatorial proof and similar proofs of other important theorems has been given by D. Zeilberger, *Discrete Math.* **56** (1985), 61–72.

Page 584 new answers ——— 17 May 1995

21. $\prod_{k=1}^{m} \binom{n_k+\cdots+n_{k-d}}{n_k}$, if we let $n_k = 0$ for $k \le 0$, since there are $\binom{n_m+\cdots+n_{m-d}}{n_m}$ ways to insert the m's into such a permutation of $\{n_1 \cdot 1, \ldots, n_{m-1} \cdot (m-1)\}$.

22. (a) The left-right reversal of $l(\pi)$ is in $P_0(0^k 1^{n_1} \ldots t^{n_t})$, for some k; but instead of reversing $l(\pi)$, we will give it a two-line form by placing 0 last instead of first in the top line. The number k of 0s in $l(\pi)$ and $r(\pi)$ is the number of columns ${j \atop k}$ in the two-line form of π for which $j \le t < k$; this is also the number of columns with $k \le t < j$. We can easily reconstruct π from the two-line forms of $l(\pi)$ and $r(\pi)$, because each column ${j \atop k}$ with $j, k \le t$ occurs in $l(\pi)$, each column with $t < j, k$ occurs in $r(\pi)$, and the remaining columns are obtained by merging ${j \atop 0}$ or ${0 \atop k}$ of $l(\pi)$ with ${0 \atop k}$ or ${j \atop 0}$ of $r(\pi)$ from left to right.

(b) Let π be a permutation of the stated form, and let σ be any permutation of $P_0(0^{n_0} 1^{n_1} \ldots m^{n_m})$. Construct λ as follows: Delete the first n_0 entries of σ; then replace the 0s by x's, subscripted with the first n_0 entries of π; replace the other elements by y's, subscripted with the remaining nonzero entries of π. Also construct ρ as follows: Delete the 0s of σ, and replace the n_j occurrences of j with x_j or y_j according as the columns ${j \atop k}$ of π have $k = 0$ or $k \ne 0$, from left to right. For example, if $\pi = {000000111112222 33333 \atop 23131302310102032010}$ and $\sigma = {000000111112222 33333 \atop 32313201103201300201}$, we have $\lambda = x_2y_2y_3x_3y_1y_1x_1y_2y_3x_3x_1y_2x_3y_1$ and $\rho = y_3y_2y_3x_1x_3x_2y_1y_1y_3y_2y_1x_3x_2x_1$. Conversely, we can reconstruct π and σ from λ and ρ.

(c) We have $w(\pi) = w\bigl(l(\pi)\bigr)\,w\bigl(r(\pi)\bigr)$ in the construction of (a), because column ${j \atop k}$ of π either becomes ${j \atop k}$ of weight w_j/w_k in $l(\pi)$ or $r(\pi)$ or it is factored into columns

${j \atop 0}$ and ${0 \atop k}$ having weights z_j/z_0 and z_0/z_k. If $l(\pi)$ has p_j columns ${0 \atop j}$ and q_j columns ${j \atop 0}$, its weight is $\prod_{j=1}^{t}(z_j^{q_j}w_j^{n_j-q_j}/z_j^{p_j}w_j^{n_j-p_j}) = \prod_{j=1}^{t}(w_j/z_j)^{p_j-q_j}$. Now $\prod_{j=1}^{t}(w_j/z_j)^{-q_j}$ is the complex conjugate of $\prod_{j=1}^{t}(w_j/z_j)^{q_j}$; so the sum of weights over all elements of $P_0(0^k1^{n_1}\ldots t^{n_t})$ simplifies to

$$\frac{k!\,(n_1+\cdots+n_t-k)!}{n_1!\ldots n_t!}\left|\sum_{p_1+\cdots+p_t=k}\binom{n_1}{p_1}\ldots\binom{n_t}{p_t}\left(\frac{w_1}{z_1}\right)^{p_1}\ldots\left(\frac{w_t}{z_t}\right)^{p_t}\right|^2.$$

Similar remarks apply to $r(\pi)$. The stated sum is positive because the term for $k=0$ is nonzero.

23. We can assume that the original strand was sorted. Let $t=2$, $m=4$, $w_1 = w_3 = z_1 = z_2 = +1$, $w_2 = w_4 = z_3 = z_4 = -1$ in part (c) of the previous exercise. Then $w(\pi) = (-1)^d$, where d is the number of columns ${j \atop k}$ with $j \neq k$. [Cf. Gillis and Zeilberger, *European J. Comb.* **4** (1983), 221–223. This result was first proved in a completely different way by Askey, Ismail, and Koornwinder, *J. Comb. Theory* **A25** (1978), 277–287, who found intriguing connections between multiset permutations and integrals of products of the Laguerre polynomials $L_n^{\alpha}(x) = \sum_{k=0}^{n}\binom{n+\alpha}{n-k}(-x)^k/k!$.] The analogous result for a five-letter alphabet is false, because the $5!$ permutations of $\{1,2,3,4,5\}$ include $1+10+45$ with an even number of differences, $0+20+44$ with an odd number.

Page 584 new answer ——————————————— 17 Aug 1995

24. (a) Transposing ${w\ x \atop y\ z}$ twice restores ${w\ x \atop y\ z}$. Given $\mathrm{sort}\binom{x_1\ \cdots\ x_n}{y_1\ \cdots\ y_n} = \binom{x_1''\ \cdots\ x_n''}{y_1''\ \cdots\ y_n''}$, unsort it by finding the leftmost x in the top row and transposing it to the left. This brings out the proper y. (The value of $\mathrm{sort}\binom{x_2'\ \cdots\ x_n'}{y_2'\ \cdots\ y_n'}$ is also uniquely determined.)

(b) We are essentially expressing the two-line notation of π in the form

$$\pi = \mathrm{sort}\begin{pmatrix} y_1 & \ldots x_{1n_1} & y_2 & \ldots x_{2n_2} & \ldots & y_t & \ldots x_{tn_t} \\ x_{11}\ldots & y_1 & x_{21}\ldots & y_2 & \ldots & x_{t1}\ldots & y_t \end{pmatrix},$$

and part (a) provides us with precisely the tools we need. [When R preserves certain statistics of the two-line notation, this construction provides combinatorial proofs of interesting theorems. Cf. Guo-Niu Han, *Advances in Math.* **105** (1994), 26–41.]

Page 585 replacement for lines 2–6 of answer 3 ——————————— 13 May 1995

$$\sum_n \frac{x^n}{n!}\sum_k \left\langle {n \atop k} \right\rangle (-1)^k = \frac{2}{e^{-2x}+1} = \frac{1}{x}\left(\frac{(-4x)}{e^{-4x}-1} - \frac{(-2x)}{e^{-2x}-1}\right)$$
$$= \frac{1}{x}\sum_{n\geq 0}\frac{B_n x^n}{n!}\big((-4)^n - (-2)^n\big);$$

hence the result is $(-1)^{n+1}B_{n+1}2^{n+1}(2^{n+1}-1)/(n+1)$. Alternatively, the identity $2/(e^{-2x}+1) = 1+\tanh x$ lets us express the answer as $(-1)^{(n-1)/2}T_n$ when n is

Page 585 answer 4 ——————————————— 13 May 1995

$\left\langle {n \atop m+1} \right\rangle \rightsquigarrow \left\langle {n \atop m} \right\rangle$

Page 585 replacement for answer 5 ——————————— 13 May 1995

5. $\left\langle {p \atop k} \right\rangle \equiv (k+1)^p - k^p \equiv (k+1) - k \equiv 1$ (modulo p) for $1 \leq k < p$, by formula (13) and Theorem 1.2.4F. Also $\left\langle {p \atop 0} \right\rangle = 1$.

Page 585 new paragraph for answer 8 ———— 25 Aug 1976

A simple $O(n^2)$ algorithm to count the number of permutations of $\{1,\ldots,n\}$ having respective run lengths $(l_1,\ldots,l_k)$ has been given by N. G. de Bruijn, *Nieuw Archief voor Wiskunde* (3) **18** (1970), 61–65.

Page 585 replacement for second sentence of answer 9 ———— 13 May 1995

Since $\sum_{k,m} q_{km} z^m x^k = \frac{x}{1-x} g(x,z)$ and $g(x,0) = 1$, we have $h(z,x) = \sum h_k(z) x^k = \frac{x}{1-x} g(x,z)(1-z^{-1}) + \frac{x}{1-x} z^{-1} = (1-z^{-1})x/(e^{(x-1)z} - x) + z^{-1}x/(1-x)$.

▸**Page 586** line 6 of answer 11 ———— 03 Oct 1997

$(-1)^k \dfrac{1}{k!} \rightsquigarrow (-1)^j \dfrac{1}{j!}$

▸**Page 587** line 1 of answer 15 ———— 03 Oct 1997

is the number $\rightsquigarrow$ is 1 plus the number

Page 588 replacement for answer 18 ———— 17 May 1995

18. (a) Let the given sequence be an inversion table as in Section 5.1.1. If it has k descents, the inverse of the corresponding permutation has k descents (see answer 5.1.1–8(e)); hence the answer is $\left\langle {n \atop k} \right\rangle$. (b) This quantity satisfies $f(n,k) = kf(n-1,k) + (n-k+1)f(n-1,k-1)$, so it must be $\left\langle {n \atop k-1} \right\rangle$. [Cf. D. Dumont, *Duke Math. J.* **41** (1974), 313–315.]

Page 588 answer 19 ———— 13 May 1995

line 1: $\left\langle {n \atop k+1} \right\rangle \rightsquigarrow \left\langle {n \atop k} \right\rangle$
line 2: $\left\langle {n \atop j+1} \right\rangle \rightsquigarrow \left\langle {n \atop j} \right\rangle$
line 4: exercise 2. $\rightsquigarrow$ exercise 2. [Chapter 5 of Riordan's *Introduction to Combinatorial Analysis* (Wiley, 1958) discusses rook placement in general.]

▸**Page 588** line −3 ———— 03 Oct 1997

$n+1-r$ readings $\rightsquigarrow$ $n+1-t$ readings

Page 589 answer 25 ———— 13 May 1995

which has k runs iff $\lfloor U_1 + \cdots + U_n \rfloor = k-1 \rightsquigarrow$ that has k descents iff $\lfloor U_1 + \cdots + U_n \rfloor = k$

Page 589 new answer ———— 07 Sep 1995

26. For example, $\vartheta^5(1-z)^{-1} = (z + 26z^2 + 66z^3 + 26z^4 + z^5)/(1-z)^6$.

Page 589 new answer ———— 28 Apr 1997

27. The following rule defines a one-to-one correspondence that takes a permutation $a_1 a_2 \ldots a_n$ with k descents into an n-node increasing forest with $k+1$ leaves: The first root is a_1, and its descendants are the forest corresponding to $a_2 \ldots a_k$, where k is minimal such that $a_{k+1} < a_1$ or $k = n$. [R. P. Stanley, *Enumerative Combinatorics* **1** (Wadsworth, 1986), Proposition 1.3.16.]

▸**Page 591** line 5 of answer 18 ———— 07 Jan 1997

$\coprod_l \rightsquigarrow \prod_l$

Page 592 answer 19 ______ 14 May 1976

(This formula ... The answer can $\leadsto$ (This formula can
$n + \frac{1}{2}\sum n_j'^2 \leadsto \frac{1}{2}(n + \sum n_j'^2)$.
also add a new paragraph:

Note: W. Feit [*Proc. Amer. Math. Soc.* **4** (1953), 740–744] showed that the number of ways to place the integers $\{1, 2, \ldots, n\}$ into an array that is the "difference" of two tableau shapes $(n_1, \ldots, n_m) \setminus (l_1, \ldots, l_m)$, where $0 \le l_j \le n_j$ and $n = \sum(n_j - l_j)$, is $n!\ \det\bigl(1/((n_j - j) - (l_i - i))!\bigr)$.

Page 592 new paragraph to follow answer 20 ______ 18 May 1995

Note: If we consider all $n!$ ways to label the nodes, the labelings considered here are those having no "inversions." Inversions in permutations are the same as inversions in tree labelings, in the special case when the tree is simply a path. See A. Björner and M. L. Wachs, *J. Combinatorial Theory* **A52** (1989), 165–187.

Page 592 and page 593, answer 22 ______ 31 May 1995

line -2 on page 592: $n > 0 \leadsto n \ge 2$
line 2 on page 593: sec $(z) \leadsto$ sec z
line 4 on page 593: $E_{2n} \leadsto |E_{2n}|$
line 5 on page 593: 688. $\leadsto$ 688; the sequence begins $(A_0, A_1, A_2, \ldots) = (1, 1, 1, 2, 5, 16, 61, 272, 1385, 7936, \ldots)$.

▸**Page 593** display in answer 25 ______ 26 Feb 1980

$z^n/n! \leadsto z^n$

▸**Page 593** bottom line ______ 24 Oct 1997

$c_i < c_{i+1}$, ... obtained. $\leadsto c_i \ge c_{i+1}$, we certainly cannot have $i + 1 \le Q_{r_i c_{i+1}} \le i$.

Page 594 new answer ______ 21 Jun 1996

28. This result is due to A. M. Vershik and S. V. Kerov, *Dokl. Akad. Nauk SSSR* **233** (1977), 1024–1028; see also B. F. Logan and L. A. Shepp, *Advances in Math.* **26** (1977), 206–222. [M. Talagrand, *Annals of Probability* **24** (1996), 1–34, Theorem 6.5, has shown that the standard deviation is $O(n^{1/4})$. Numerical computations by A. Odlyzko and E. Rains in 1994 suggest that it actually is proportional to $n^{1/6}$.]

Page 594 new answers ______ 30 Apr 1996

32. Let $m_n = \int_{-\infty}^{\infty} t^n e^{-(t-1)^2/2} dt/\sqrt{2\pi}$. Then $m_0 = m_1 = 1$, and $m_{n+1} - m_n = n m_{n-1}$ if we integrate by parts. So $m_n = t_n$ by (40).

33. True; it is $\det_{i,j=1}^m \binom{n_i}{j-1}$. [Mitchell, in *Amer. J. Math.* **4** (1881), 341–344, showed that it is the number of terms in the expansion of a certain symmetric function, now called a Schur function.]

Page 594 new answer ______ 19 May 1995

34. The length of a hook is also the length of any zigzag path from the hook's bottom left cell (x, y) to its top right cell (x', y'). We prove a stronger result: If there's a hook of length $a + b$, then there's either a hook of length a or a hook of length b. Consider the cells $(x, y) = (x_1, y_1)$, (x_2, y_2), ..., $(x_{a+b}, y_{a+b}) = (x', y')$ that hug the bottom of the shape. If $x_{a+1} = x_a$, the cell (x_a, y_1) has a hook of length a; otherwise (x_{a+1}, y_{a+b}) has a hook of length b. [*Reference: Japanese J. Math.* **17** (1940), 165–184, 411–423. Nakayama was the first to consider hooks in the study of permutation groups, and he came close to discovering Theorem H.]

Page 594 new answers ________ 20 May 1995

35. The execution of steps G3–G5 decreases exactly h_{ij} elements of the p array by 1 when q_{ij} is increased, because the algorithm follows a zigzag path from $p_{c_j j}$ to p_{ir_i}. The next execution of those steps either starts with a larger value of j or stays above or equal to the preceding zigzag. Therefore the q array is filled from left to right and bottom to top; to reverse the process we proceed from right to left and top to bottom:

H1. [Initialize.] Set $p_{ij} \leftarrow 0$ for $1 \le j \le r_i$ and $1 \le i \le c_1$. Then set $i \leftarrow 1$ and $j \leftarrow r_1$.

H2. [Find nonzero cell.] If $q_{ij} > 0$, go on to step H3. Otherwise if $i < c_j$, increase i by 1 and repeat this step. Otherwise if $j > 1$, decrease j by 1, set $i \leftarrow 1$, and repeat this step. Otherwise stop (the q array is now zero).

H3. [Decrease q, prepare for zigzag.] Decrease q_{ij} by 1 and set $l \leftarrow i$, $k \leftarrow r_i$.

H4. [Move down or left.] If $l < c_k$ and $p_{lk} > p_{(l+1)k}$, increase l by 1 and return to H4. Otherwise if $k > j$, decrease k by 1 and return to H4. Otherwise return to H2. ▌

The first zigzag path for a given column j ends by incrementing $p_{c_j j}$, because $p_{1j} \le \cdots \le p_{c_j j}$ implies that $p_{c_j j} > 0$. Each subsequent path for column j stays below or equal to the previous one, so it also ends at $p_{c_j j}$. The inequalities encountered on the way show that this algorithm inverts the other. [*J. Combinatorial Theory* **A21** (1976), 216–221.]

36. (a) The stated coefficient of z^m is the number of solutions to $m = \sum h_{ij} q_{ij}$, so we can apply the result of the previous exercise. (b) If $a_1, \ldots, a_k$ are any positive integers, we can prove by induction on k that

$$[z^m]\,1/(1-z)(1-z^{a_1})\ldots(1-z^{a_k}) = \binom{m}{k}\Big/a_1\ldots a_k + O(m^{k-1})\,.$$

The number of partitions of m with at most n parts is therefore $\binom{m}{n-1}/n! + O(m^{n-2})$ for fixed n, by exercise 5.1.1–15. This is also the asymptotic number of partitions $m = p_1 + \cdots + p_n$ with *distinct* parts $p_1 > \cdots > p_n > 0$ (see exercise 5.1.1–16). So the number of reverse plane partitions is asymptotically $N\binom{m}{n-1}/n! + O(m^{n-2})$ when there are N tableaux of a given n-cell shape. By part (a) this is also $\binom{m}{n-1}/\prod h_{ij} + O(m^{n-2})$. [*Studies in Applied Math.* **50** (1971), 167–188, 259–279.]

37. Plane partitions in a rectangle are equivalent to reverse plane partitions, so the hook lengths tell us the generating function $1/\prod_{i=1}^{r}\prod_{j=1}^{c}(1-z^{i+j-1})$ in an $r \times c$ rectangle. Letting $r, c \to \infty$ yields $1/(1-z)(1-z^2)^2(1-z^3)^3\ldots$. [MacMahon's original derivation in *Philosophical Transactions* **211** (1912), 75–110, 345–373, was extremely complicated. The first reasonably simple proof was found by Carlitz, *Acta Arithmetica* **13** (1967), 29–47.]

Page 594 new answer ________ 21 May 1995

38. (a) The probability is $1/n$ when $k = l = 1$; otherwise it is

$$\frac{nP(I\setminus\{i_0\}, J) + nP(I, J\setminus\{j_0\})}{n\,d_{i_0 j_0}} = \frac{(d_{i_0 b} + d_{a j_0})/(n\,d_{i_0 b}\ldots d_{i_{k-1} b}\,d_{a j_0}\ldots d_{a j_{l-1}})}{d_{i_0 b} + d_{a j_0}}\,,$$

by induction on $k + l$.

(b) Summing over all I and J gives

$$n^{-1}(1+d_{1b}^{-1})\ldots(1+d_{(a-1)b}^{-1})\,(1+d_{a1}^{-1})\ldots(1+d_{a(b-1)}^{-1})\,,$$

which is easily seen to equal $f(T\setminus\{(a,b)\})/f(T)$.

(c) The sum over all corners yields 1, because every path ends at a corner. Therefore $\sum f(T\setminus\{(a,b)\}) = f(T)$, and this proves Theorem H by induction on n. Furthermore, if we put n into the corner cell at the end of the random path and repeat the process on the remaining $n-1$ cells, we get each tableau with probability $1/f(T)$. [*Advances in Math.* **31** (1979), 104–109.]

Page 594 new answer ______ 16 May 1996

39. An equivalent process has been analyzed by H. Rost, *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete* **58** (1981), 41–53.

Page 594 new answers ______ 07 Sep 1996

Move the former answer 39 from page 605 to this page, changing it to number 40, and making the following correction to the final line: *Math.*, to appear]. $\leadsto$ *Math.* **11** (1975), 29–35].)

41. Construct a multigraph that has vertices $\{0_R, 1_L, 1_R, \ldots, n_L, n_R, (n+1)_L\}$ and edges k_R — $(k+1)_L$ for $0\le k\le n$; also include the edges 0_R — 7_R, 7_L — 1_L, 1_R — 2_L, 2_R — 4_L, 4_R — 5_L, 5_R — 3_L, 3_R — 6_R, 6_L — 8_L, which define the "bonds" of *Lobelia fervens*. Exactly two edges touch each vertex, so the connected components are cycles: $(0_R\,1_L\,7_L\,6_R\,3_R\,4_L\,2_R\,3_L\,5_R\,6_L\,8_L\,7_R)(1_R\,2_L)(4_R\,5_L)$. Any flip operation changes the number of cycles by -1, 0, or $+1$. Therefore we need at least five flips to reach the eight cycles $(0_R\,1_L)(1_R\,2_L)\ldots(7_R\,8_L)$. [J. Kececioglu and D. Sankoff, *Lecture Notes in Comp. Sci.* **807** (1994), 307–325.]

The first flip must break the bond 6_L — 8_L, because we get no new cycle when we break two bonds that have the same left-to-right orientation in the linear arrangement. This leaves five possibilities after one flip, namely $g_7^R g_6 g_3^R g_5^R g_4^R g_2^R g_1^R$, $g_7^R g_1 g_2 g_4 g_5 g_3 g_6$, $g_7^R g_1 g_2 g_6 g_3^R g_5^R g_4^R$, $g_7^R g_1 g_2 g_4 g_5 g_6 g_3^R$, and $g_6 g_3^R g_5^R g_4^R g_2^R g_1^R g_7$; four more flips suffice to sort all but the second of these.

Incidentally, there are $2^7\cdot 7! = 645120$ different possible arrangements of $g_1\ldots g_7$, and 179904 of them are at distance ≤ 5 from tobacco order.

[An efficient algorithm to find the best way to sort any signed permutation by reversals was developed by S. Hannenhalli and P. Pevzner, *STOC* **27** (1995), 178–189, and improved to run in $O(n^2)$ time by Kaplan, Shamir, and Tarjan, *SODA* **8** (1997), 344–351.]

42. Denote an arrangement like $g_7^R g_1 g_2 g_4 g_5 g_3 g_6^R$ by the signed permutation $\overline{7}12453\overline{6}$. If there is a negated element, say $\overline{k}$ is present but not $\overline{k-1}$, one flip will create the 2-cycle $((k-1)_R\,k_L)$. Similarly, if $\overline{k}$ is present but not $\overline{k+1}$, a single flip creates $(k_R\,(k+1)_L)$. And if all flips of that special kind remove all negated elements, a single flip creates two 2-cycles. If no negated elements are present and the permutation isn't sorted, some flip will preserve the number of cycles. Hence we can sort in $\le n$ flips if the given permutation has a negated element, $\le n+1$ otherwise.

When n is even, the permutation $n\,(n-1)\,\ldots\,1$ requires $n+1$ flips, because it has one cycle after the first flip. When $n>3$ is odd, the permutation $2\,1\,3\,n\,(n-1)\,\ldots\,4$ requires $n+1$ by a similar argument.

43. Let c_k be the number of cycles of length $2k$ in the multigraph of the previous answers. An upper bound on the average value of c_k can be found as follows: The total number of potential $2k$-cycles is $2^k(n+1)^{\underline{k}}/(2k)$, because we can choose a sequence of k distinct edges from $\{0_R - 1_L, \ldots, n_R - (n+1)_L\}$ in $(n+1)^{\underline{k}}$ ways and orient them in 2^k ways; this counts each cycle $2k$ times, including impossible cases like $(1_R\,2_L\,2_R\,3_L)$ or $(1_R\,2_L\,3_L\,2_R\,3_R\,4_L)$ or $(1_R\,2_L\,6_R\,7_L\,4_L\,3_R\,2_R\,3_L\,6_L\,5_R)$. When $k \le n$, every possible $2k$-cycle occurs in exactly $2^{n-k}(n-k)!$ signed permutations. For example, consider the case $k=5$, $n=9$, and the cycle $(0_R\,1_L\,9_L\,8_R\,7_R\,8_L\,1_R\,2_L\,5_L\,4_R)$. This cycle occurs in the multigraph if and only if the signed permutation begins with $\overline{4}$ and contains the substrings $\overline{9}18\overline{7}$ and $\overline{2}5$ or their reverses; we obtain all solutions by finding all signed permutations of $\{1,2,3,6\}$ and replacing 1 by $\overline{9}18\overline{7}$, 2 by $\overline{2}5$. Therefore $\mathrm{E}\,c_k \le \frac{1}{2k}2^k(n+1)^{\underline{k}}2^{n-k}(n-k)!/2^n n! = \frac{1}{2}\bigl(\frac{1}{k}+\frac{1}{n+1-k}\bigr)$. It follows that $\mathrm{E}\,c = \sum_{k=1}^{n}\mathrm{E}\,c_k + \mathrm{E}\,c_{n+1} < H_n + 1$. Since $n+1-c$ is a lower bound on the number of flips, we need $\ge n+1-\mathrm{E}\,c > n - H_n$ of them.

[This proof uses ideas of V. Bafna and P. Pevzner, *SICOMP* **25** (1996), 272–289, who studied the more difficult problem of sorting *unsigned* permutations by reversals. In that problem, an interesting permutation that can be written as the product of non-disjoint cycles $(1\,2\,3)(3\,4\,5)(5\,6\,7)\ldots$, ending with either $(n{-}1\,n)$ or $(n{-}2\,n{-}1\,n)$ depending on whether n is even or odd, turns out to be the hardest to sort.]

Page 599 replacement for line 3 of answer 14 ———————— 31 May 1995

$$A_{2n} = \sum_{0\le i\le j}(j-i)\binom{i+j}{i}\binom{2n-i-j}{n-j} = \sum_{i,k\ge 0} k\binom{2i+k}{i}\binom{2n-2i-k}{n-i-k};$$

Page 599 answer 14 line 7 ———————— 29 Nov 1977

13); $\rightsquigarrow$ 13), and still another by the identity in the answer to exercise 5.2.2–16 with $f(k)=k$;

Page 599 replacement for last line of answer 15 ———————— 14 May 1976

with $n+1$ nodes; cf. exercise 2.3.4.5–5. It is interesting to note that $G(w,z) = F(-wz,z)/F(-w,z)$, where $F(z,q) = \sum_{n\ge 0} z^n q^{n^2}/\prod_{k=1}^{n}(1-q^k)$ is the generating function for partitions $p_1+\cdots+p_n$ into n parts subject to the conditions $p_j \ge p_{j+1}+2$ for $1\le j<n$ and $p_n>0$ (cf. exercise 5.1.1–16).

▶**Page 600** answer 18 ———————— 11 Jun 1985

line 2: $4h_{s+1}^2 h_{s-1} \rightsquigarrow 4h_{s-1}^2 h_{s+1}$
line 4: $N^{1+(2^{t-1}+1)/(2^t-1)} \rightsquigarrow N^{1+2^{t-1}/(2^t-1)}$

Page 600 lines 1–5 of answer 21 ———————— 30 May 1995

If $ah+bk$ ... this argument $\rightsquigarrow$ If $xh+yk = x'h+y'k$, we have $(x-x')h = (y'-y)k$, so $x' = x+tk$ and $y' = y-th$ for some integer t. Let $h'h+k'k=1$; then $n = (nh')h+(nk')k$, so every integer n has a unique representation of the form $n = xh+yk$ where $0\le x<k$, and n is generable iff $y\ge 0$. Let, similarly, $hk-h-k-n = x'h+y'k$; then $(x+x')h+(y+y')k = hk-h-k$. Hence $x+x' \equiv k-1$ (modulo k) and we must have $x+x' = k-1$. Hence $y+y' = -1$, and $y\ge 0$ iff $y'<0$.

The symmetry of this result

Page 601 last line of answer 24 ———————— 28 May 1995

1972). $\rightsquigarrow$ 1972). Heuristics that find permutations needing even more moves are discussed by H. Erkiö, *BIT* **20** (1980), 130–136. See also Weiss and Sedgewick, *J. Algorithms* **11** (1990), 242–251, for improvements on Pratt's construction.

Page 601 replacement for bottom three lines ——— 29 May 1995

A random 3- and 2-ordered permutation therefore has approximately $\frac{n}{5}(\phi^{-1} + 2\phi^{-2}) = n\phi^{-1}/\sqrt{5} \approx .276n$ inversions. But if a random permutation is 3-sorted, then 2-sorted, exercise 40 shows that it has $\approx n/4$ inversions; if it is 2-sorted, then 3-sorted, it has $\approx n/3$.

Page 602 new answer ——— 29 May 1995

27. (a) See *J. Algorithms* **15** (1993), 101–124. A simpler proof, which shows that c can be any constant $< \frac{1}{2}$, was found independently by C. G. Plaxton and T. Suel, *FOCS* **33** (1992), 226–235. (b) This is obvious if $m > \frac{1}{4}c^2(\ln N/\ln\ln N)^2$. Otherwise $N^{1+c/\sqrt{m}} \geq N(\ln N)^2$. R. E. Cypher [*SICOMP* **22** (1993), 62–71] has proved the slightly stronger bound $\Omega\bigl(N(\log N)^2/\log\log N\bigr)$ when the increments satisfy $h_{s+1} > h_s$ for all s.

Page 602 answer 28 ——— 12 Oct 1996

255 633 15 7 3 1 $\leadsto$ 209 109 41 19 5 1

Page 602 answer 29 ——— 12 Oct 1996

line 2: 7100 $\leadsto$ 7210
line 2: 8300 $\leadsto$ 8170
line 3: of $126457u$ $\leadsto$ $\approx 127720u$
line 3: $131002u$ $\leadsto$ $\approx 128593u$
line 4: increments (8) $\leadsto$ increments (11)
line 5: $N^{(s-1)/t}$ $\leadsto$ $N^{s/t}$
line 8: 7600). $\leadsto$ 7950). Its average running time, $\approx 127260u$, is the best known so far.
line 10: 17600 $\leadsto$ 18240
line 11: 11500 $\leadsto$ 11865

▸**Page 602** line *03* of the program ——— 12 Feb 1976

`ENN2 INPUT-N,4` $\leadsto$ `ENN2 -INPUT-N,4`

Page 603 answer 33, comments to program ——— 31 Jul 1976

line *07*: $p \equiv \mathrm{rI2}$, $q \equiv \mathrm{rI3}$ $\leadsto$ $p \equiv \mathrm{rI3}$, $q \equiv \mathrm{rI2}$
lines *09* and *15*: To L4 $\leadsto$ To L4 with $q \leftrightarrow p$

Page 604 replacement copy for lines 3 and 4 ——— 14 Jun 1977

The ∞ trick also speeds up Program S; the following code suggested by J. H. Halperin uses this idea and the `MOVE` instruction to reduce the running time to $(6B + 11N - 10)u$, assuming that location `INPUT+N+1` already contains the largest possible one-word value:

```
START ENT2 N-1          1
2H    LDA  INPUT,2      N-1
      ENT1 INPUT,2      N-1
      JMP  3F           N-1
4H    MOVE 1,1(1)       B
3H    CMPA 1,1          B+N-1
      JG   4B           B+N-1
5H    STA  0,1          N-1
      DEC2 1            N-1
```

10 `J2P 2B` $N-1$

Doubling up the inner loop would save an additional $B/2$ or so units of time.

Page 604 replacement for lines 8 and following of answer 37 ———— 05 May 1996

If $G_M(w,z)$ is the stated double generating function, differentiation by z gives

$$G'_M(w,z) = M\left(\sum_{n\ge0} g_n(z)\frac{w^n}{n!}\right)^{M-1} \sum_{n\ge0} g'_n(z)\frac{w^n}{n!},$$

hence

$$\sum_{N\ge0} g'_{MN}(1)\frac{M^N w^N}{N!} = Me^{(M-1)w}\left(\frac{w^2}{4}e^w\right) = \frac{M}{4}w^2e^{Mw};$$

similarly, the formula $g''_n(1) = \frac{3}{2}\binom{n}{4} + \frac{5}{3}\binom{n}{3}$ yields

$$\sum_{N\ge0} g''_{MN}(1)\frac{M^N w^N}{N!} = M(M-1)e^{(M-2)w}\left(\frac{w^2}{4}e^w\right)^2 + Me^{(M-1)w}\left(\frac{3}{2}w^4 + \frac{5}{3}w^3\right)e^w.$$

Equating coefficients of w^N tells us that $g'_{NM}(1) = \frac{1}{2}\binom{N}{2}M^{-1}$, $g''_{NM}(1) = \left(\frac{3}{2}\binom{N}{4} + \frac{5}{3}\binom{N}{3}\right)M^{-2}$, and the variance is $\left(\frac{1}{6}\binom{N}{3} + \frac{2M-1}{4}\binom{N}{2}\right)M^{-2}$.

Page 605 replacement for answer 39, which moved to page 594 ———— 28 May 1995

39. We have $a_k = O(\rho^k)$, because the prime number theorem implies that the number of primes between ρ^k and ρ^{k+1} is $\rho^{k+1}/(k+1) - \rho^k/k + O(\rho^k/k^2)$; this is positive for all sufficiently large k. Therefore the sum of the first $\binom{k}{2}$ elements of (10) is $\sum_{1\le i<j\le k} f(a_i,a_j) = \sum_{1\le i<j\le k} O(\rho^{i+j})$; and we have

$$\sum_{1\le i<j\le k} \rho^{i+j} = \frac{\rho^3(\rho^k-1)(\rho^{k-1}-1)}{(\rho^2-1)(\rho-1)}.$$

(b) If $\binom{k-1}{2} < \log_\rho N \le \binom{k}{2}$ we have $(k-1)^2 < 2\log_\rho N$, hence $\rho^{2k} = O(\exp c\sqrt{\ln N}\,)$.

Notice that as $\rho \to 1$, the base sequence $a_1, a_2, \ldots$ becomes equal to the sequence of prime numbers, and the bound in Theorem I reduces to $O\bigl(N(\log N)^4(\log\log N)^{-3}\bigr)$.

Page 605 new answer ———— 30 May 1995

40. (a) [A. C. Yao, *J. Algorithms* **1** (1980), 14–50.] We can show that each of the $\binom{h}{2}$ pairs of lists contributes $\frac{\sqrt{\pi}}{4}g^{-2}h^{-3/2}N^{3/2} + O(N/gh)$ inversions to each subfile $(K_a, K_{a+g}, K_{a+2g}, \ldots)$, $1 \le a \le g$. For example, suppose $h = 12$, $g = 5$, $a = 1$, and consider inversions where the lists $K_3 < K_{15} < K_{27} < \cdots$ and $K_7 < K_{19} < K_{31} < \cdots$ intersect the subfile $(K_1, K_6, K_{11}, \ldots)$. After the first pass, $(K_3, K_7, K_{15}, K_{19}, K_{27}, K_{31}, \ldots)$ is a random 2-ordered permutation. The elements K_j of concern to us have $j \equiv 1$ (modulo 5) and $j \equiv 3$ or 7 (modulo 12); hence $j \equiv 51$ or 31 (modulo 60), and we want to compute the average value of $g(51,31)$ where

$$g(x,y) = \sum_{j<k}\bigl([K_{x+ghj} > K_{y+ghk}] + [K_{y+ghj} > K_{x+ghk}]\bigr) + r(x,y),$$

$$r(x,y) = \sum_j [K_{\min(x,y)+ghj} > K_{\max(x,y)+ghj}] < N/gh.$$

If $|p| \le g$ and $|q| \le g$ we have

$$[K_{j+ph-gh} > K_{k+qh+gh}] \le [K_j > K_k] \le [K_{j+ph+gh} > K_{k+qh-gh}]\,;$$

hence

$$\begin{aligned}&[K_{x+ghj} > K_{y+ghk}] + [K_{y+ghj} > K_{x+ghk}]\\ &\qquad \le [K_{x+ph+gh(j+1)} > K_{y+qh+gh(k-1)}] + [K_{y+qh+gh(j+1)} > K_{x+ph+gh(k-1)}]\end{aligned}$$

and it follows that $g(x,y) \le g(x+ph, y+qh) + 8N/gh$. Similarly we find $g(x,y) \ge g(x+ph, y+qh) - 8N/gh$. But the sum of $g(x,y)$ over all g^2 pairs (x,y) such that $x \bmod h = b$ and $y \bmod h = c$, for any given $b \ne c$, is the total number of inversions in a random 2-ordered permutation of $2N/h$ elements. Therefore by exercise 14, the average value of $g(x,y)$ is $g^{-2}\sqrt{\pi/128}\,(2N/h)^{3/2} + O(N/gh)$.

(b) See S. Janson and D. E. Knuth, *Random Structures and Algorithms* **10** (1997), 125–142. When h and g are large we have $\psi(h,g) = \sqrt{\pi h/128}\,g + O(g^{-1/2}h^{1/2}) + O(gh^{-1/2})$.

Page 605 addendum to answer 2 ——— 14 May 1976

(Algorithm 5.2.3S does exactly $\mathrm{xch}(\pi)$ exchanges; see exercise 5.2.3–4.)

▸**Page 606** answer 11 ——— 29 Nov 1977

In general, ... elements. $\leadsto$ The situation becomes more complicated when N is larger; in general, the set $\{K_2, K_4, \ldots\}$ should be $\{1, 2, \ldots, M-1, M, M+2, M+4, \ldots, 2\lfloor N/2\rfloor - M\}$, permuted so as to maximize the exchanges for $\lfloor N/2\rfloor$ elements. Here $M = \lceil 2^k/3\rceil$, where k maximizes $k\lfloor N/2\rfloor - \frac{1}{9}\big((3k-2)2^{k-1} + (-1)^k\big)$. The maximum total number of exchanges is $1 - 2\lg\lg N/\lg N + O(1/\log N)$ times the number of comparisons [R. Sedgewick, *SICOMP* **7** (1978), 239–272].

Page 606 replacement for answer 12 ——— 03 Mar 1986

12. The following program by W. Panny avoids the `AND` instruction by noting that step M4 is performed for $i = r + 2kp + s$, $k \ge 0$, and $0 \le s < p$. Here $\mathtt{TT} \equiv 2^{t-1}$, $p \equiv \mathrm{rI1}$, $r \equiv \mathrm{rI2}$, $i \equiv \mathrm{rI3}$, $i + d - N \equiv \mathrm{rI4}$, and $p - 1 - s \equiv \mathrm{rI5}$; we assume that $N \ge 2$.

`START`	`ENT1`	`TT`	1	M1. Initialize p. $p \leftarrow 2^{t-1}$.
`2H`	`ENT2`	`TT`	T	M2. Initialize q, r, d.
	`ST2`	`Q(1:2)`	T	$q \leftarrow 2^{t-1}$.
	`ENT2`	`0`	T	$r \leftarrow 0$.
	`ENT4`	`0,1`	T	$\mathrm{rI4} \leftarrow d$.
`3H`	`ENT3`	`0,2`	A	M3. Loop on i. $i \leftarrow r$.
	`INC4`	`-N,3`	A	$\mathrm{rI4} \leftarrow i + d - N$.
`8H`	`ENT5`	`-1,1`	$D+E$	$s \leftarrow 0$.
`4H`	`LDA`	`INPUT+1,3`	C	M4. Compare/exchange.
	`CMPA`	`INPUT+N+1,4`	C	
	`JLE`	`*+4`	C	Jump if $K_{i+1} \le K_{i+d+1}$.
	`LDX`	`INPUT+N+1,4`	B	
	`STX`	`INPUT+1,3`	B	$R_{i+1} \leftrightarrow R_{i+d+1}$.
	`STA`	`INPUT+N+1,4`	B	
	`J5Z`	`7F`	C	Jump if $s = p - 1$.

```
        DEC5 1          C − D   s ← s + 1.
        INC3 1          C − D   i ← i + 1.
        INC4 1          C − D
        J4N  4B         C − D   Repeat loop if i + d < N.
        JMP  5F           E     Otherwise go to M5.
   7H   INC3 1,1          D     i ← i + p + 1.
        INC4 1,1          D
        J4N  4B           D     Repeat loop if i + d < N.
   5H   ENT2 0,1          A     M5. Loop on q. r ← p.
   Q    ENT4 *            A     rI4 ← q.
        ENTA 0,4          A
        SRB  1            A
        STA  Q(1:2)       A     q ← q/2.
        DEC4 0,1          A     rI4 ← d.
        J4P  3B           A     To M3 if d ≠ 0.
   6H   ENTA 0,1          T     M6. Loop on p.
        SRB  1            T
        STA  *+1(1:2)     T
        ENT1 *            T     p ← ⌊p/2⌋.
        J1NZ 2B           T     To M2 if p ≠ 0.
```

The running time depends on six quantities, only one of which depends on the input data (the remaining five are functions of N alone): $T = t$, the number of "major cycles"; $A = t(t+1)/2$, the number of passes or "minor cycles"; $B =$ the (variable) number of exchanges; $C =$ the number of comparisons; $D =$ the number of blocks of consecutive comparisons; and $E =$ the number of incomplete blocks. When $N = 2^t$, it is not difficult to prove that $D = (t-2)N + t + 2$ and $E = 0$. For Table 1, we have $T = 4$, $A = 10$, $B = 3+0+1+4+0+0+8+0+4+5 = 25$, $C = 63$, $D = 38$, $E = 0$, so the total running time is $11A + 6B + 10C + 2E + 12T + 1 = 939u$.

In general when $N = 2^{e_1} + \cdots + 2^{e_r}$, Panny has shown that $D = e_1(N+1) - 2(2^{e_1} - 1)$, $E = \binom{e_1 - e_r}{2} + (e_1 + e_2 + \cdots + e_{r-1}) - (e_1 - 1)(r-1)$.

Page 607 replacement for answer 16 ______ 29 Nov 1977

16. Consider the $\binom{2n}{n}$ lattice paths from $(0,0)$ to (n,n) as in Figs. 11 and 18, and attach weight $f(i-j)$ if $i \ge j$, $f(j-i-1)+1$ if $i < j$, to the line from (i,j) to $(i+1,j)$; here $f(k)$ is the number of bit variations $b_r \ne b_{r+1}$ in the binary expansion $k = (\ldots b_2 b_1 b_0)_2$. The total number of exchanges on the final merge when $N = 2n$ is then $\sum_{0\le j\le i<n} (2f(j)+1)\binom{2i-j}{i-j}\binom{2n-2i+j-1}{n-i-1}$. R. Sedgewick showed that this sum simplifies, for general f, to $\frac{n}{2}\binom{2n}{n} + 2\sum_{k\ge 1}\binom{2n}{n-k}\sum_{0\le j<k} f(j)$; then he used the gamma function method to obtain the asymptotic formula

$$\binom{2n}{n}\left(\frac{1}{4}n\lg n + \left(\lg\frac{\Gamma(1/4)^2}{2\pi} + \frac{1}{4} + \frac{\gamma+2}{4\ln 2} + \delta(n)\right)n + O(\sqrt{n}\log n)\right),$$

where $\delta(n)$ is a periodic function of $\lg n$ with magnitude bounded by .0005. Hence about 1/4 of the comparisons lead to exchanges, on the average, as $n \to \infty$. [*SICOMP* **7** (1978), 239–272; see also Flajolet and Odlyzko, *SIAM J. Discrete Math.* **3** (1990), 238–239.]

▸**Page 608** replacement for answer 19 —— 13 Feb 1986

19. Yes, the other subfiles may be processed in any order. But the queue will contain $\Omega(N/\sqrt{\log N}\,)$ items when each partitioning step divides the file equally, while a stack is guaranteed to stay much smaller than this (see the next exercise).

▸**Page 608** answer 20 —— 10 Jan 1984

$\lfloor \lg(N+2)/(M+2)\rfloor \rightsquigarrow \max\bigl(0, \lfloor \lg(N+2)/(M+2)\rfloor\bigr)$

▸**Page 608** display for $C_N(z)$ —— 15 Mar 1979

$z^{N+1-\delta_{s1}} \rightsquigarrow z^{N+1}$

▸**Page 608** line −5 —— 08 Jan 1997

$S_{n-s}(z).\ \rightsquigarrow S_{n-s}(z),$

▸**Page 609** publisher's mixup in answers 24 and 29 —— 01 Mar 1979

line 3 of answer 24: replace by lines 8 and 9 of answer 29
lines 8 and 9 of answer 29 should be:

$\alpha \neq \beta$; $g(x) = x^{\beta}(\ln x + C)$ for $\alpha = \beta$. We have $p_t(-t-2) = 0$; so the general solution to our differential equation is

▸**Page 610** second line of answer 31 —— 29 Nov 1977

step $\rightsquigarrow$ stop

Page 610 append a new paragraph to answer 31 —— 11 Jul 1996

R. G. Dromey [*Software Practice & Experience* **16** (1986), 981–986] has observed that fewer comparisons and exchanges are needed if we stop each partitioning stage as soon as i or j has reached position m.

Page 611 append a new paragraph to answer 32 —— 22 Jul 1996

Another solution follows from the theory of Section 6.2.2: Suppose the keys are $\{1, 2, \ldots, n\}$, and let X_{jk} be the number of common ancestors of nodes j and k in the binary search tree corresponding to quicksort. Then the number of comparisons made by the algorithm of exercise 31 can be shown to be $\sum_{j=1}^{n} X_{jm} + X_{mm} - 2[\text{node } m$ is a leaf$]$. The probability that node i is a common ancestor of nodes j and k in a random binary search tree is $1/\bigl(\max\{i,j,k\} - \min\{i,j,k\} + 1\bigr)$. We obtain the average number of comparisons from the facts that $\mathrm{E}\,X_{jk} = H_j + H_{n+1-k} + 1 - 2H_{k-j+1}$ for $1 \le j \le k$, and Pr(node m is a leaf) = Pr(m isn't followed by $m \pm 1$ in a random permutation) $= \frac{1}{3} + \frac{1}{6}\delta_{m1} + \frac{1}{6}\delta_{mn} + \frac{1}{3}\delta_{m1}\delta_{mn}$. [Cf. R. Raman, *SIGACT News* **25**, 2 (June 1994), 86–89.]

▸**Page 611** the bottom two (clobbered) lines should start respectively thus: 14 May 1976

43. As $a \to 0+$,
$\Gamma(1))/a \to \Gamma'(1) = -\gamma$,
[This correction has already been made, in sufficiently recent printings.]

▸**Page 612** answer 44 —— 19 Feb 1986

line 1: $\Gamma\bigl((k+1)/2\bigr) \rightsquigarrow \Gamma\bigl((k+1)/2\bigr) - \delta_{k0}$
line 9: $-\frac{1}{6} \rightsquigarrow +\frac{1}{3}$

▸**Page 612** answer 48 line 4 —— 28 Feb 1978

$\int_{1/2-i\infty}^{1/2+i\infty} \rightsquigarrow \int_{-1/2-i\infty}^{-1/2+i\infty}$

▸**Page 613** line 1 ——— 10 Jan 1984

$(-1)^{n+1} \rightsquigarrow (-1)^n$

▸**Page 613** line −3 ——— 29 Oct 1997

$1.521 \rightsquigarrow 1.504$

▸**Page 614** line 8 of answer 56 ——— 30 Oct 1997

$\ldots\ (m+2)\ +\ \bigl((m+1)\ldots \rightsquigarrow \ldots\ (m+2)\ -\ \bigl((m+1)\ldots$

▸**Page 614** line 11 of answer 56 ——— 30 Oct 1997

$\frac{12}{7}\ (p+1) \rightsquigarrow \frac{12}{7}/(p+1)$

Page 615 line 4 ——— 23 Jun 1996

$2.11N.$ $\rightsquigarrow$ $2.116N$ [*Acta Inf.* **7** (1977), 336–341].

Page 615 new answer ——— 14 Sep 1996

58. This is a special case of a general theorem due to R. M. Karp; see *JACM* **41** (1994), 1136–1150, §2.8.

Page 615 answer 1 ——— 27 Jun 1996

No; but $\rightsquigarrow$ No; consider the case $K_1 > K_2 = \cdots = K_N$. But

Page 616 replacement for answer 7 ——— 23 May 1995

7. See A. C. Yao, "On straight selection sort," Computer Science Technical Report 185 (Princeton University, 1988).

Page 617 replacements for lines −6 and −5 ——— 14 Dec 1979

Eratosthenes, as implemented in exercise 4.5.4–8, is a $O(N \log\log N)$ method requiring considerably more random access storage. More efficient implementations are discussed in Section 7.1.

Page 618 answer 18 ——— 21 Oct 1996

line 7: $K_j \leftarrow K_i \rightsquigarrow R_j \leftarrow R_i$
line 10: An instruction $\rightsquigarrow$ A `MIX` instruction
new paragraph to follow line 10: [For further analysis of this idea, see I. Wegener, *Theoretical Comp. Sci.* **118** (1993), 81–98.]

▸**Page 619** answer 27 ——— 27 May 1978

$\sum_{d\backslash n} \rightsquigarrow \sum_{d\backslash N}$

Page 619 new paragraph for the end of answer 28 ——— 15 Dec 1996

For further information about t-ary heaps, see S. Okoma, *Lecture Notes in Comp. Sci.* **88** (1980), 439–451.

Page 619 new answer ——— 19 Oct 1996

30. Suppose $n = 2^t - 1 + r$, where $t = \lfloor \lg n \rfloor$ and $1 \le r \le 2^t$. Then $h_{2m} = [m=0]$ and

$$h_{(n+1)m} \le \sum_{j=0}^{t-2} (2^j - 1) h_{n(m-j)} + 2^{t-1} h_{n(m-t+1)} + r h_{n(m-t)} \qquad \text{for } n \ge 2,$$

by considering the number of elements on level j that could be the final resting place of K_{n+1} after it has been sifted up in place of K_1. Therefore, if $g_{nm} = h_{nm}/2^m$, we have

$$g_{(n+1)m} \le \sum_{j=0}^{t-1} \frac{2^j - 1}{2^j} g_{n(m-j)} + g_{n(m-t+1)} + \frac{r}{2^t} g_{n(m-t)} \le \bigl(\lg(n+1)\bigr) \max_{m\ge 0} g_{nm},$$

and it follows by induction that $g_{nm} \le L_n = \prod_{k=2}^{n} \lg k$.

The average total number of promotions during the selection phase is $B''_N = h_N^{-1} \sum_{m\ge 0} m h_{Nm}$, where $h_N = \sum_{m\ge 0} h_{Nm}$ is the total number of possible heaps (Theorem H). We know that $B''_N \le N\lceil \lg N\rceil$. On the other hand, we have $B''_N \ge m - h_N^{-1} \sum_{k=1}^{m} (m-k) h_{Nk} \ge m - h_N^{-1} L_N \sum_{k=1}^{m} (m-k) 2^k > m - 2^{m+1} h_n^{-1} L_N$, for all m. Choosing $m = \lg(h_N/L_N) + O(1)$ now gives $B''_N \ge \lg(h_N/L_N) + O(1)$.

The number of comparisons needed to create a heap is at most $2N$, by exercise 23(b); hence $h_N \ge N!/2^{2N}$. Clearly $L_N \le (\lg N)^N$, so we have $\lg(h_N/L_N) \ge N\lg N - N\lg\lg N + O(N)$. [*J. Algorithms* **15** (1993), 76–100.]

Page 619 answer 31 ——— 16 Mar 1981

lines 1 and 2: Let ... $B[i]$ for ⇝ (Solution by J. Edighoffer, 1981.) Let A be an array of $2n$ elements such that $A[2\lfloor i/2\rfloor] \le A[2i]$ and $A[2\lfloor i/2\rfloor - 1] \ge A[2i-1]$ for $1 < i \le n$; furthermore we require that $A[2i-1] \ge A[2i]$ for
line 4: twin-heap ⇝ twin heap

Page 619 last line of answer 31 ——— 17 Oct 1996

details. ⇝ details. This idea was independently discovered and developed further by J. van Leeuwen and D. Wood [*Comp. J.* **36** (1993), 209–216], who called the structure an "interval heap."

Page 620 replacement for answer 34 ——— 15 Mar 1995

34. Starting with the recurrence

$$L_1(z) = z, \qquad L_{m+1}(z) = L_m(z)\left(L(z) - \sum_{k=1}^{m-1} L_k(z)\right),$$

for parts of the overall generating function $L(z) = \sum_{n\ge 0} l_n z^n = \sum_{m\ge 1} L_m(z)$, where $L_m(z) = z^{2^{m-1}} + \cdots$ generates leftist trees with shortest path length m from root to leaf, Rainer Kemp has proved that $L(z) = z + \frac{1}{2}L(z)^2 + \frac{1}{2}\sum_{m\ge 1} L_m(z)^2$, and that $a \approx 0.25036$ and $b \approx 2.7494879$ [*Inf. Proc. Letters* **25** (1987), 227–232]. Luis Trabb Pardo noticed about 1980 that the generating function $G(z) = zL(z)$ satisfies the elegant relation $G(z) = z + G\bigl(zG(z)\bigr)$.

Page 620 new answer ——— 19 Oct 1996

38. In any heap of N distinct elements, the largest $m = \lceil N/2\rceil$ elements form a subtree. At least $\lfloor m/2\rfloor$ of them must be nonleaves of that subtree, since a binary tree with k leaves has at least $k-1$ nonleaves. Therefore at least $\lfloor m/2\rfloor$ of the largest m elements appear in the first $\lfloor N/2\rfloor$ positions of the heap. Those elements must be promoted to the root position before reaching their final destinations; so their movement contributes $\ge \sum_{k=1}^{\lfloor m/2\rfloor} \lfloor \lg k\rfloor = \frac{1}{2} m\lg m + O(m)$ to B, by exercise 1.2.4–42. Thus $B_{\min}(N) \ge \frac{1}{4}N\lg N + O(N) + B_{\min}(\lfloor N/2\rfloor)$, and the result follows by induction on N. [I. Wegener, *Theoretical Comp. Sci.* **118** (1993), 81–98, Theorem 5.1. Schaffer and Sedgewick, and independently Bollobás, Fenner, and Frieze, have constructed permutations that require no more than $\frac{1}{2}N \lg N + O(N \log\log N)$ promotions; see *J. Algorithms* **15** (1993), 76–100; **20** (1996), 205–217. Such permutations are quite rare, by the result of exercise 30.]

Page 620 new answer ________________ 29 Dec 1996

39. In an infinite heap, the kth-largest element is equally likely to appear in the left or the right subheap of its larger ancestors. Thus we can use the theory of digital search trees, obtaining $e(k) = \overline{C}_k - \overline{C}_{k-1}$ in the notation of Eq. 6.3–(13). By exercise 6.3–28 we have $e(k) = \lg k + \gamma/(\ln 2) + \frac{1}{2} - \alpha_2 + f_0(k) + O(k^{-1}) \approx \lg k - .274$, where $f_0(k)$ is a periodic function of $\log k$, and $\alpha_2 \approx 1.6067$. [P. V. Poblete, *BIT* **33** (1993), 411–412.]

▸**Page 623** answer 18 ________________ 23 Jun 1996

line 1: simplest known method uses $\leadsto$ first solution to be found used
line 2: [*Dokladi* $\leadsto$ [cf. *Doklady*
line 9: Z_1. $\leadsto$ Z_1; if more than one zone has the smallest leading element, choose one that has smallest trailing element. (This
line 10: element, $\leadsto$ and trailing elements,
line 13: has less $\leadsto$ will have fewer
new paragraph for the end: See answer 5.5–3 for the problem of *stable* merging in place.

▸**Page 624** line −5 ________________ 01 May 1979

$g_{M,N}^{n+1}(z) \leadsto g_{M,N}^{(n+1)}(z)$

Page 626 new answer ________________ 31 May 1995

16. Assume that each string is followed by a special null character that is less than any letter of the alphabet. Perform a left-to-right radix sort by starting with all strings linked together in a single block of data. Then for $k = 1, 2, \ldots,$ refine every block that contains more than one distinct string by splitting it into subblocks based on the kth letter of each string, meanwhile keeping the blocks sorted by their already-examined prefixes. When a block has only one item, or when its kth characters are all null (so that its keys are identical), we can arrange to avoid examining it again. [R. Paige and R. E. Tarjan, *SICOMP* **16** (1987), 973–989, §2.] This process is essentially that of constructing a trie as in Section 6.3. A simpler but slightly less efficient algorithm based on right-to-left radix sort was given for this problem by Aho, Hopcroft, and Ullman, *The Design and Analysis of Computer Algorithms* (Addison-Wesley, 1974), 79–84.

▸**Page 626** line 1 of answer 1 ________________ 10 Mar 1997

Interchange left and right subtrees for all internal nodes of the trees for A_{ij}.

Page 627 lines 12 and 13 ________________ 13 May 1995

2^{k-1} times ... $\sum_k \left\langle {n \atop k} \right\rangle 2^{k-1}$. $\leadsto$ 2^k times if the permutation $a_1 \ldots a_n$ contains k ascents; hence P_n can be expressed in terms of the Eulerian numbers, $P_n = \sum_k \left\langle {n \atop k} \right\rangle 2^k$.

Page 627 line 17 ________________ 05 Jan 1979

J. Touchard $\leadsto$ P. A. MacMahon, *Proc. London Math. Soc.* **22** (1891), 341–344; J. Touchard

Page 627 bottom of page, new paragraph for answer 6 ________________ 25 Aug 1976

M. Paterson observes that if the multiplicities of keys are $(n_1, \ldots, n_m)$, the number of comparisons can be reduced to $n \lg n - \sum n_j \lg n_j + O(n)$; see *SICOMP* **5** (1976), 2.

▸**Page 630** lines 5–7 ——— 27 May 1978

level $l-1$ ... $\lfloor \lg N \rfloor + 1$; $\leadsto$ level l and $N-k$ on level $l+1$, where $0 < k \le N$. By exercise 2.3.4.5–3, $k2^{-l} + (N-k)2^{-l-1} = 1$; hence $N + k = 2^{l+1}$. The inequalities $2^l \le N < 2^{l+1}$ now show that $l = \lfloor \lg N \rfloor$;

Page 631 new answer ——— 05 May 1996

35. [*Mat. Zametki* **4** (1968), 511–518.] See S. Felsner and W. T. Trotter, *Combinatorics, Paul Erdős is Eighty* **1** (1993), 145–157, for a summary of progress on this problem, and for a proof that we can always achieve $1 \le T(G_1)/T(G_2) < 8/3 - \epsilon$ for sufficiently small $\epsilon > 0$.

Page 632 new answer ——— 06 May 1996

4. No. [C. Christen, *FOCS* **19** (1978), 259–266.]

▸**Page 633** answer 13 ——— 14 May 1976

397–404 $\leadsto$ 263–269

Page 633 new answer ——— 11 Nov 1988

14. [*SICOMP* **9** (1980), 298–320. The complete solution for $M(4,n)$ was obtained shortly afterwards by J. Schulte Mönting, who also gave a conjectured solution for $M(5,n)$, in *Theor. Comp. Sci.* **14** (1981), 19–37.]

Page 633 new answer ——— 23 May 1995

22. R. Michael Tanner [*SICOMP* **7** (1978), 18–38] has shown that a "fractile insertion" algorithm makes at most $1.06 \lg \binom{m+n}{m}$ comparisons on the average.

Page 633 lines −8, −7, −6, −3; page 634 lines 2, 17; page 637 line 2 ——— 14 May 1976

oracle $\leadsto$ adversary

Page 634 new answer ——— 17 Oct 1996

24. The general problem is as hard as the special case where each x_{ij} is 0 or 1 and $x = \frac{1}{2}$. Then each comparison is equivalent to looking at the bit x_{ij}, and we want to determine the entire matrix by inspecting the fewest bits. Any merging problem (1) corresponds to such a 0-1 matrix if we set $x_{ij} = [A_i > B_{n+1-j}]$. (N. Linial and M. Saks, in *J. Algorithms* **6** (1985), 86–103, attribute this observation to J. Shearer. A similar result connects searching and sorting with respect to any partial order.)

Page 634 replacement for answer 4 ——— 25 Aug 1996

4. Suppose the largest $t-1$ elements are $\{a_1, \ldots, a_{t-1}\}$. Any path in the comparison tree to determine the largest t elements, consistent with this assumption, must include at least $n-t$ comparisons to determine the largest of the remaining $n-t+1$ elements. Such paths have at least $n-t$ binary choice points, so there are at least 2^{n-t} of them. Thus, each of the $n^{\underline{t-1}}$ choices for the largest $t-1$ elements must appear in at least 2^{n-t} leaves of the tree.

▸**Page 634** answer 6 line 1 ——— 29 Nov 1977

$\lg(2^{l_1} + 2^{l_2} + \cdots + 2^{l_m}) \leadsto \lceil \lg(2^{l_1} + 2^{l_2} + \cdots + 2^{l_m}) \rceil$

Page 635 answer 9 ——— 14 May 1976

comparisons.) $\leadsto$ comparisons, yet the procedure is not optimum for $\overline{V}_3(5)$.)

▸**Page 635** diagram for answer 9 ______ 30 Sep 1982

replace the leaves of the tree, and the spurious numbers below those leaves, by

$$\boxed{3}\quad\boxed{2}\quad\boxed{5}\quad\boxed{1}\quad\boxed{5}\quad\boxed{2}\quad\boxed{3}\quad\boxed{1}$$

Page 636 answer 14 ______ 14 May 1976

line 1: found in $\leadsto$ found in $U_t(n) \le$
also add a new sentence after line 2: (Kirkpatrick actually proved that (12) is a lower bound for $U_t(n+1)-1$. For larger t, an improved bound for $U_t(n)$ was found by J. W. John, *SICOMP* **17** (1988), 640–647.)

Page 637 replacement for answer 21 ______ 25 Aug 1996

21. [*J. Algorithms* **5** (1984), 557–578.]

Step 0. Build two knockout trees of sizes 2^k and 2^{k-t+1}.

Step j, for $1 \le j \le t$. (At this point we have output the largest $j-1$ elements. The remaining elements, together with a set of dummy placeholders that each equal $-\infty$, now appear in two knockout trees A and B, where A has 2^k leaves and B has 2^{k-t+j}.) Let a be the champion of A, and assume that a has beaten a_0, a_1, …, a_{k-1}, where a_l is a champion of 2^l elements. Similarly, let b and b_0, b_1, …, $b_{k-t+j-1}$ be the champion and subchampions of B. If $j=t$, output $\max(a,b)$ and stop. Otherwise, "grow" another level at the bottom of B by introducing 2^{k-t+j} dummies who each have lost their first game to the players of B. (Our strategy will be to merge B into A, if possible, by exchanging it with the subtree A' of A that contains a_0, a_1, …, a_{k-t+j}; notice that A', like the newly enlarged B, is a knockout tree with $2^{k-t+j+1}$ leaves.) Compare b to $a_{k-t+j+1}$, then compare the winner to $a_{k-t+j+2}$, etc., until $c = \max(b, a_{k-t+j-1}, \ldots, a_{k-1})$ has been found. Case 1, $b<c$: Output a and interchange B with A'. Case 2, $b=c$ and $b<a$: Output a and interchange B with A'. Case 3, $b=c$ and $b>a$: Output b. After handling these three cases we are left with (possibly new) knockout trees A and B in which the champion of B has just been output. Remove that element from B and replace it by $-\infty$, making any necessary comparisons to restore the knockout tournament structure (as in tree selection). This completes Step j.

Step 0 makes $2^k-1+2^{k+1-t}-1$ comparisons, and Step t makes 1. Steps 1, 2, …, $t-1$ each make at most $k-1$ comparisons, except in Case 2 when there might be k. But whenever Case 2 occurs, we'll save one comparison the next time we're in Case 1 or Case 2, because a_0 will then be $-\infty$. Thus steps 1 through $t-1$ make at most $(t-1)(k-1)+1$ comparisons altogether.

By exercise 3 we have $W_t(n) \le n+(t-1)(k-1)$ for all $n \le 2^k+2^{k+1-t}$, when $k \ge t \ge 2$. If $n \ge 2^k+t-2$, exercise 4 says that $W_t(n) \ge n-t+\lceil \lg (2^k+t-2)^{\underline{t-1}}\rceil$, which is $n-t+(t-1)k+1$ if $t \ge 3$. Thus the method is optimum for $2^k+t-2 \le n \le 2^k+2^{k+1-t}$ when $k \ge t \ge 3$. (Also for several smaller values of n, if t is large.)

A similar method, which uses a reserved element instead of $-\infty$ when rebuilding B at the end of steps 1, …, $t-2$ (see the proof of (11)), proves that $V_t(n) \le n+(t-1)(k-1)$ when $n \le 2^k+2^{k+1-t}+t-2$ and $k \ge t \ge 3$.

Page 637 new answer ______ 14 May 1976

22. In general when $2^r \cdot 2^k < n+2-t \le (2^r+1)\cdot 2^k$ and $t < 2^r \le 2t$, this procedure starting with $t+1$ knockout trees of size 2^k will yield $\lfloor (t-1)/2 \rfloor$ fewer comparisons than (11), since at least this many of the comparisons used to find the minimum in (ii) can be "reused" in (iii).

Page 637 new answer ———————————————— 14 Feb 1996

23. According to (15), the quantity $V_{\lceil n/2\rceil}(n)/n$ is bounded below by 2 as $n \to \infty$; but D. Dor [Ph.D. thesis, Tel Aviv University, 1995] has shown that the actual lower limit is strictly greater than 2. Dor and U. Zwick have also established the upper limit 2.942 [*SODA* **6** (1995), 28–37].

Page 637 new answer ———————————————— 13 Sep 1996

24. Since $W_t(n) = n + O(t\log n)$ by Eq. (6), the statement in the hint is surely true when $t \le \sqrt{n/\ln n}$. Suppose that statement holds for n, and let u and v have ranks $t_- = \lfloor t-\sqrt{t\ln n}\,\rfloor$ and $t_+ = \lceil t+\sqrt{t\ln n}\,\rceil$ in the first n of $2n$ randomly ordered elements. (The smallest element has rank 1.) Compare the other n elements to v, and compare those less than v also to u. The probability p_s that an element x of rank t in the first n has rank s overall is $\binom{s-1}{t-1}\binom{2n-s}{n-t}/\binom{2n}{n}$. The average value of s is $\sum sp_s = \frac{2n+1}{n+1}t$; this is the average number of elements $< x$, hence the average number of comparisons to u is $\binom{n}{n+1}t_+ = t + O(n\log n)^{1/2}$. Let u and v have ranks s_- and s_+ among all $2n$ elements, and let $T_- = \lfloor 2t - \sqrt{2t\ln 2n}\,\rfloor$, $T_+ = \lceil 2t + \sqrt{2t\ln 2n}\rceil$. If $s_- \le T_-$ and $s_+ \ge T_+$, we can find the elements of ranks T_- and T_+ by selecting from the $s_+ - s_- + 1$ elements between u and v. We will prove that it's very unlikely to have $s_- > T_-$ or $s_- < T_- - 2\sqrt{n\ln n}$ or $s_+ < T_+$ or $s_+ > T_+ + 2\sqrt{n\ln n}$; therefore $O(n\log n)^{1/2}$ further comparisons will almost always suffice. The hint will follow by induction on n if we can show that "very unlikely" means "with probability $O(n^{-1-\epsilon})$ for all sufficiently large n."

Notice that $p_{s+1}/p_s = s(n-s+t)/(s+1-t)(2n-s)$ decreases as s increases from t to $n+t$, and it is ≤ 1 if and only if $s \ge 2n(t-1)/(n-1)$; it is $\le 1 - \frac{1}{2}cn^{-1/2} + O(n^{-1})$ when $s = \overline{s}(c) = 2t + ct(n-t)/n^{3/2}$. Therefore the probability that $s \ge \overline{s}(c)$ is $\le 2c^{-1}n^{1/2}p_{\overline{s}(c)}\bigl(1+O(n^{-1/2})\bigr)$. Similarly, $p_{s-1}/p_s < 1 - \frac{1}{2}cn^{-1/2} - O(n^{-1})$ when $s = \underline{s}(c) = 2t - 1 - c(t-1)(n+1-t)/n^{3/2}$, so $s \le \underline{s}(c)$ with probability $\le 2c^{-1}n^{1/2}p_{\underline{s}(c)}\bigl(1+O(n^{-1/2})\bigr)$. In the cases we need, the relevant values of c are $\ge .55n^{3/2}(\ln n)^{1/2}t^{-1/2}(n-t)^{-1}$ for all large n, and Stirling's approximation implies that $p_{\overline{s}(c)}$ and $p_{\underline{s}(c)}$ are both

$$\begin{aligned} O\bigl(n^{1/2}s^{-1/2}(2n-s)^{-1/2}\bigr)&\exp(-2sc^2(n-t)^2/n^3 - 2(2n-s)c^2t^2/n^3)\\ &\le O\bigl(t^{-1/2}\exp(-4t(n-t)c^2/n^2)\bigr) \le O\bigl(t^{-1/2}n^{-1.2}\bigr). \end{aligned}$$

Thus the probability $O\bigl(n^{-1.2}(\log n)^{1/2}\bigr)$ is indeed very unlikely. [A similar construction appeared in *CACM* **18** (1975), 165–172, but the analysis was incorrect.]

Page 637 new answer ———————————————— 11 Sep 1996

25. Given a selection algorithm and a permutation π of $\{1,\ldots,n\}$, let's charge each comparison $\pi_i : \pi_j$ to π_i if $|\pi_i - t| > |\pi_j - t|$; if $|\pi_i - t| = |\pi_j - t|$, we charge $\frac{1}{2}$ to each. A charge to π_i is called *useful* if $\pi_i < \pi_j \le t$ or $\pi_i > \pi_j \ge t$; otherwise it's *useless.* Let x_k be the total charge to k. Then the total number of comparisons is $x_1 + \cdots + x_n$. Clearly $x_t = 0$; but $x_k \ge 1$ for all $k \ne t$, because every element other than t has a useful charge. We will prove that $\mathrm{E}\,x_{t+k} + \mathrm{E}\,x_{t-k} \ge 3$ for $0 < k < t$.

Let $A_k(\pi) = [$the first charge to $t+k$ was useless$]$. Then $A_k(\pi) = 1 - A_{-k}(\pi')$, where π' is like π but with the elements $(t-k,\ldots,t+k-1,t+k)$ replaced respectively by $(t-k+1,\ldots,t+k,t-k)$. Therefore $\mathrm{E}\,A_k + \mathrm{E}\,A_{-k} = 1$.

Let $B_k(\pi) = [$the first charge to both $t+k$ and $t-k$ was $\frac{1}{2}$, and $t+k$ received its second charge before $t-k$ did$]$. Also let $C_k(\pi) = [x_{t+k} \ge 2 + A_k]$. Then $B_k(\pi) \le$

$C_k(\pi')$, where π' is like π but with the elements $(t-k, t-k+1, \ldots, t+k-1)$ replaced by $(t+k-1, t-k, \ldots, t+k-2)$. Similarly, $B_{-k}(\pi) \le C_{-k}(\pi'')$, where π'' is obtained from π by changing $(t-k+1, \ldots, t+k-1, t+k)$ to $(t-k+2, \ldots, t+k, t-k+1)$. It follows that $\mathrm{E}\,B_k \le \mathrm{E}\,C_k$ and $\mathrm{E}\,B_{-k} \le \mathrm{E}\,C_{-k}$.

The proof is completed by observing that $x_{t-k}+x_{t+k} \ge 2+A_k+A_{-k}-B_k-B_{-k}+C_k+C_{-k}$. [See *JACM* **36** (1989), 270–279, for further results.]

The upper bound in (17) also has a matching lower bound: Andrew and Frances Yao proved that $\overline{V}_t(n) \ge n+\frac{1}{2}t\,(\ln\ln n - \ln t - 9)$ for $t > 1$ and $n \ge (8t)^{18t}$, in *SICOMP* **11** (1982), 428–447.

Page 637 new answer ______ 19 Dec 1976

26. (a) Let the vertices of the two types of components be designated a; $b < c$. The adversary acts as follows on nonredundant comparisons: Case 1, $a : a'$, make an arbitrary decision. Case 2, $x : b$, say that $x > b$; all future comparisons $y : b$ with this particular b will result in $y > b$, otherwise the comparisons are decided by an adversary for $U_t(n-1)$, yielding $\ge 2+U_t(n-1)$ comparisons in all. This reduction will be abbreviated "let $b = \min$; $2+U_t(n-1)$." Case 3, $x : c$, let $c = \max$; $2+U_{t-1}(n-1)$.

(b) Let the new types of vertices be designated $d_1, d_2 < e$; $f < g < h > i$. Case 1, $a : a'$ or $c : c'$, arbitrary decision. Case 2, $a : c$, say that $a < c$. Case 3, $x : b$, let $b = \min$; $2+U_t(n-1)$. Case 4, $x : d$, let $d = \min$; $2+U_t(n-1)$. Case 5, $x : e$, let $e = \max$; $3+U_{t-1}(n-1)$. Case 6, $x : f$, let $f = \min$; $2+U_t(n-1)$. Case 7, $x : g$, let f and $g = \min$; $3+U_t(n-2)$. Case 8, $x : h$, let $h = \max$; $3+U_{t-1}(n-1)$. Case 9, $x : i$, let $i = \min$; $2+U_t(n-1)$.

(c) For $t = 1$ we have $U_t(n) = n-1$, so the inequality holds. For $1 < t \le n/2-1$, use induction and (b). For $t = (n-1)/2$, use induction and (a). For $t = n/2$, $U_t(n-1) = U_{t-1}(n-1)$; use induction and (a).

Page 637 new answer ______ 14 Feb 1995

27. (a) The height h satisfies $2^h \ge \sum_l 1 \ge \sum_l \Pr(l)/p = 1/p$.

(b) If $r \le t$, we reach A3 after at least $n-|S_0|-|T_0| = n-|S_0|-r$ flips. The tth largest element will be either the smallest or largest element of Q, and the elements of Q have not yet been compared to each other, so we will need at least $|Q|-1$ more flips. If $|S_0| < q$ we have $|Q| = r$, and if not we have $|Q| \ge |S_0|-|C(y_0)|+1 \ge |S_0|-(q-r)+1$; so in both cases at least $n-q$ flips will be made. There are $n+1-t$ sets T containing the $t-1$ largest elements determined by a given leaf, and for every such T the probability of reaching that leaf is either zero or $2^{-f}/\binom{n}{t}$, where $f \ge n-q$ is the number of flips corresponding to T. [This adversary is implicit in the paper of Bent and John, *STOC* **17** (1985), 213–216.]

(c) If $t < r$, change t to $n+1-t$; this will make $t \ge r$ when r maximizes the right-hand side, since r will be $O(\sqrt{n}\,)$. If it is possible to reach A3 with $|C(y)| > q-r$ for all $y \in T_0$, the algorithm will make $n-1$ comparisons to relate the tth largest element to all the others, in addition to at least $(r-1)(q-r+1)$ comparisons it made between S and $T \setminus \{y_0\}$.

(d) Choose $r = \lceil \sqrt{m}\,\rceil$ and $q = 2r-2$. (It is slightly better to let $q = r + \lfloor \sqrt{m} + \frac{1}{2} \rfloor - 2$; this choice maximizes the lower bound derived in (c).)

[Improved upper bounds, coming close to (15) when α is small, have been obtained by Dor and Zwick, *Combinatorica* **16** (1996), 41–58.]

Page 637 end of answer 1 — 07 May 1996

diagram.) ⇝ diagram. This change is valid because the swapped lines are being compared to each other.)

Page 637 line 3 of answer 4 — 23 May 1995

$\widehat{T}(7) = \widehat{T}(8) = 6$.] ⇝ $\widehat{T}(7) = \widehat{T}(8) = 6$. Ian Parberry has shown by exhaustive computer search that $\widehat{T}(9) = \widehat{T}(10) = 7$; see *Math. Systems Theory* **24** (1991), 101–116.]

Page 638 new sentence for end of answer 9 — 21 Oct 1996

Similarly $\widehat{M}(3,4) = 8$. But what are $\widehat{M}(3,5)$ and $\widehat{M}(4,5)$?

▸**Page 638** line 1 of answer 12 — 09 Sep 1997

$\langle 6,7,8,2\rangle$ ⇝ $\langle 8,7,6,2\rangle$

Page 639 last line of answer 26 — 29 Jul 1996

6.5–1. ⇝ 6.5–1; see also exercise 51.

Page 640 answer 36 last line — 14 May 1976

to appear ⇝ 333–339

▸**Page 640** answer 37 line 5 — 11 Nov 1997

$\pi^{n/2}$ ⇝ π^n

Page 640 answer 37 line 8 — 30 Sep 1996

Kautz et al. ⇝ Kautz, Levitt, and Waksman

Page 640 amendment to answer 38 — 14 May 1976

G. Baudet and D. Stevenson have observed that exercises 37 and 38 combine to yield a simple sorting method with $(n \ln n)/k + O(n)$ comparison cycles on k processors: First sort k subfiles of size $\le \lceil n/k \rceil$, then merge them in k passes using the "odd-even transposition merge" of order k. [*IEEE Trans.* **C-27** (1978), 84–87.]

Page 641 answer 43 — 02 Mar 1977

A. Waksman and M. Green have proved that ⇝ By slightly extending a construction due to L. J. Goldstein and S. W. Leibholz, *IEEE Trans.* **EC-16** (1967), 637–641, one can show that $P(n) \le P(\lfloor n/2 \rfloor) + P(\lceil n/2 \rceil) + n - 1$, hence
Eq. 5.3.1–3 ... Green also ⇝ Eq. 5.3.1–(3); M. W. Green

Page 641 new answer — 23 May 1995

44. Ajtai, Komlós, and Szemerédi have shown that $O(1)$ levels of m-sorters will sort m^2 elements. [To appear.]

Page 641 new answer — 07 Feb 1996

46. [*J. Algorithms* **3** (1982), 79–88; the following alternative proof is due to V. S. Grinberg.] We may assume that $1 \le m \le n$ and that every stage makes m comparisons. Let $l = \lceil (n-m)/2 \rceil$ and suppose we are merging $x_1 \le \cdots \le x_m$ with $y_1 \le \cdots \le y_n$. An adversary can force $\lceil \lg(m+n) \rceil$ stages as follows: In the first stage some x_j is compared to an element y_k where we have either $k \le l$ or $k \ge l+m$. The adversary decides that $x_{j-1} < y_1$ and $x_{j+1} > y_n$; also that $x_j > y_k$ if $k \le l$, $x_j < y_k$ if $k \ge l+m$. The remaining task is essentially to merge x_j with either $y_{k+1} \le \cdots \le y_n$ or $y_1 \le \cdots \le y_{k-1}$; so at least $\min(n-k+1, k) \ge \min(n-l+1, l+m) = \lceil (m+n)/2 \rceil$ outcomes remain. At least $\lceil \lg \lceil (m+n)/2 \rceil \rceil = \lceil \lg(m+n) \rceil - 1$ subsequent stages are therefore necessary.

Page 641 new answer ____ 23 May 1995

47. C. G. Plaxton and T. Suel have shown that any such network requires at least $\Omega\bigl((\log n)^2/\log\log n\bigr)$ levels of delay. [*Math. Systems Theory* **27** (1994), 491–508.]

Page 641 line -2 ____ 16 Jan 1977

Pollard.] $\rightsquigarrow$ Pollard.] All such identities can be obtained from a system of four axioms and a rule of inference for multivalued logic due to Łukasiewicz; see Rose and Rosser, *Trans. Amer. Math. Soc.* **87** (1958), 1–53.

Page 641 new answer ____ 31 Oct 1996

49. [Cf. H. Zhu and R. Sedgewick, *STOC* **14** (1982), 296–302.] The stated delay time is easily verified by induction. But the problem of analyzing the recurrence

$$A(m,n) = A\bigl(\lfloor m/2\rfloor, \lceil n/2\rceil\bigr) + A\bigl(\lceil m/2\rceil, \lfloor n/2\rfloor\bigr) + \lceil m/2\rceil + \lceil n/2\rceil - 1,$$

when $A(0,n) = A(m,0) = 0$, is more difficult.

A bitonic merge makes $B(m,n) = C'(m+n)$ comparisons. Therefore we can use the fact that $\bigl\{\lfloor m/2\rfloor + \lceil n/2\rceil, \lceil m/2\rceil + \lfloor n/2\rfloor\bigr\} = \bigl\{\lfloor (m+n)/2\rfloor, \lceil (m+n)/2\rceil\bigr\}$ to show that $B(m,n) = B\bigl(\lfloor m/2\rfloor, \lceil n/2\rceil\bigr) + B\bigl(\lceil m/2\rceil, \lfloor n/2\rfloor\bigr) + \lfloor (m+n)/2\rfloor$. Then $A(m,n) \le B(m,n)$ by induction.

Let $D(m,n) = C(m+1,n+1) + C(m,n) - C(m+1,n) - C(m,n+1)$. We have $D(0,n) = D(m,0) = 1$, and $D(m,n) = 1$ when $m+n$ is odd. Otherwise $m+n$ is even and $mn \ge 1$, and we have $D(m,n) = D\bigl(\lfloor m/2\rfloor, \lfloor n/2\rfloor\bigr) - 1$. Consequently $D(m,n) \le 1$ for all $m, n \ge 0$.

The recurrence for A is equivalent to the recurrence for C except when m and n are both odd. And in that case we have $A(m,n) \ge C\bigl(\lfloor m/2\rfloor, \lceil n/2\rceil\bigr) + C\bigl(\lceil m/2\rceil, \lfloor n/2\rfloor\bigr) + \lceil m/2\rceil + \lceil n/2\rceil - 1 = C(m,n) + 1 - D\bigl(\lfloor m/2\rfloor, \lfloor n/2\rfloor\bigr) \ge C(m,n)$ by induction.

Let $l = \lceil \lg\min(m,n)\rceil$. On level k of the even-odd recursion, for $0 \le k < l$, we perform 2^k merges of the respective sizes $(m_{jk}, n_{jk}) = \bigl(\lfloor (m+j)/2^k\rfloor, \lfloor (n+2^k-1-j)/2^k\rfloor\bigr)$ for $0 \le j < 2^k$. The cost of recursion, $\sum_j \bigl(\lceil m_{jk}/2\rceil + \lceil n_{jk}/2\rceil - 1\bigr)$, is $f_k(m) + f_k(n) - 2^k$; we can write $f_k(n) = \max(n'_k, n - n'_k)$, where $n'_k = 2^k\lfloor n/2^{k+1} + 1/2\rfloor$ is the multiple of 2^k that is nearest to $n/2$. Since $0 \le f_k(n) - n/2 \le 2^{k-1}$, the total cost of recursion for levels 0 to $l-1$ lies between $\frac12(m+n)l - 2^l$ and $\frac12(m+n)l$.

Finally, if $m \le n$, the 2^l merges (m_{jl}, n_{jl}) on level l have $m_{jl} = 0$ for $0 \le j < 2^l - m$, and $m_{jl} = 1$ for the other m values of j. Since $A(1,n) = n$, the total cost of level l is $\sum_{k=n}^{m+n-1} \lfloor k/2^l\rfloor \le \sum_{k=n}^{m+n-1} k/m = \frac{m-1}{2} + n$.

Thus even-odd merging, unlike bitonic merging, is within $O(m+n)$ of the optimum number of comparisons $\widehat{M}(m,n)$. Our derivation shows in fact that $A(m,n) = \sum_{k=0}^{l-1}\bigl(f_k(m) + f_k(n) - 2^k\bigr) + g_l(m+n) - g_l(\max(m,n))$, where $g_l(n)$ can be expressed in the form $\sum_{k=0}^{n-1}\lfloor k/2^l\rfloor = \lfloor n/2^l\rfloor\bigl(n - 2^{l-1}(\lfloor n/2^l\rfloor + 1)\bigr)$.

Page 641 new answer ____ 29 Jul 1996

51. In fact we can construct α_x inductively so that $x\alpha_x = 0^{k-1}101^{n-k-1}$, when x has k zeros. The base case, α_{10}, is empty. Otherwise at least one of the following four cases applies, where y is not sorted: (1) $x = y0$, $\alpha_x = \alpha_y[n-1:n][n-2:n-1]\ldots[1:2]$. (2) $x = y1$, $\alpha_x = \alpha_y[1:n][2:n]\ldots[n-1:n]$. (3) $x = 0y$, $\alpha_x = \alpha_y^+[1:n][1:n-1]\ldots[1:2]$. (4) $x = 1y$, $\alpha_x = \alpha_y^+[1:2][2:3]\ldots[n-1:n]$. [Cf. M. J. Chung and B. Ravikumar, *Discrete Math.* **81** (1990), 1–9.] This construction uses $\binom{n}{2} - 1$ comparators; can it be done with substantially fewer?

▸**Page 642** line 14 ______ 27 May 1978

$K_{h[1]+j} \leftarrow K_{h[m]+j} \rightsquigarrow K_{h[1]+j} > K_{h[m]+j}$

Page 643 new answer ______ 13 May 1996

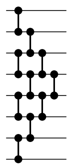
61. (Gil Kalai has informally announced a proof of minimality for the restricted case, using methods related to his article in *Graphs and Combinatorics* **1** (1985), 65–79; however, his proof has not yet been published.)

Page 643 last line of answer 62 ______ 30 Apr 1996

sorting.] $\rightsquigarrow$ sorting. These portions of his thesis have been reprinted in *IEEE Transactions* **C34** (1985), 296–310.]

Page 643 new answer ______ 14 May 1996

63. When $n = 8$, for example, such a network must include the comparators shown here; all other comparators are ineffective on 10101010. Then lines $\lceil n/3\rceil \mathinner{.\,.} \lceil 2n/3\rceil = 3 \mathinner{.\,.} 6$ sort 4 elements. (This exercise is based on an idea of David B. Wilson.)



Notes: There is a one-to-one correspondence between minimal-length primitive networks that sort a given bit string and Young tableaux whose shape is bounded by the zigzag path defined by that bit string. Thus, exercise 51 gives a one-to-one correspondence between primitive networks of $\binom{n/2+1}{2}$ comparators that sort $(10)^{n/2}$ and primitive networks of $\binom{n/2+1}{2}$ comparators that sort $n/2+1$ arbitrary numbers. If a primitive network sorts the bit string $1^{n/2}0^{n/2}$, we can make a stronger statement: All of its "halves," consisting of the subnetworks on lines k through $k + n/2$ inclusive, are sorting networks, for $1 \le k \le n/2$.

Page 643 new answer ______ 16 May 1996

64. This follows by applying the tail inequalities to the interesting construction in Proposition 7 of a paper by H. Rost, *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete* **58** (1981), 41–53, setting $b = \frac{1}{2}$, $a = \frac{1}{4}$, and $t = 4n$.

Curiously, R. P. Stanley and S. V. Fomin have proved that if the comparators $[i_k : i_k + 1]$ are chosen nonuniformly in such a way that $i_k = j$ occurs with probability $j/\binom{n}{2}$, then the expected time to reach a primitive sorting network (not necessarily bubble sort) is exactly $\binom{n}{2}H_{\binom{n}{2}}$.

Page 643 new answer ______ 24 May 1995

66. Insert the elements $i_1, \ldots, i_N$ into an initially empty tableau using Algorithm 5.1.4I but with one crucial change: Set $P_{ij} \leftarrow x_i$ in step I3 only if $x_i \ne P_{i(j-1)}$. It can be proved that x_i will equal $P_{i(j-1)}$ in that step only if $x_i + 1 = P_{ij}$, when the inputs $i_1 \ldots i_N$ define a primitive sorting network. (The parenthesized assertions of the algorithm need to be modified.) After i_j has been inserted into P, set $Q_{st} \leftarrow j$ as in Theorem 5.1.4A. After N steps, the tableau P will always contain $(r, r+1, \ldots, n-1)$ in row r, while Q will be a tableau from which the sequence $i_1 \ldots i_N$ can be reconstructed by working backwards.

For example, when $n = 6$ the sequence $i_1 \ldots i_N = 4\,1\,3\,2\,4\,3\,5\,4\,3\,1\,2\,3\,5\,1\,4$ corresponds to

$$P = \begin{array}{|c|c|c|c|c|}\hline 1&2&3&4&5\\\hline 2&3&4&5\\\cline{1-4} 3&4&5\\\cline{1-3} 4&5\\\cline{1-2} 5\\\cline{1-1}\end{array}\,, \qquad Q = \begin{array}{|c|c|c|c|c|}\hline 1&4&5&8&13\\\hline 2&6&7&15\\\cline{1-4} 3&9&12\\\cline{1-3} 10&11\\\cline{1-2} 14\\\cline{1-1}\end{array}\,.$$

The transpose of Q corresponds to the complementary network $[n-i_1 : n-i_1+1] \ldots [n-i_N : n-i_N+1]$.

References: A. Lascoux and M. P. Schützenberger, *Comptes Rendus Acad. Sci.*, Série I, **295** (Paris, 1982), 629–633; R. P. Stanley, *Eur. J. Combinatorics* **5** (1984), 359–372; P. H. Edelman and C. Greene, *Advances in Math.* **63** (1987), 42–99. The diagrams of primitive sorting networks also correspond to arrangements of pseudolines and to other abstractions of two-dimensional convexity; see D. E. Knuth, *Lecture Notes in Comp. Sci.* **606** (1992), for further information.

Page 645 new paragraph after answer 10 ———— 28 Feb 1978

Purists might complain that the algorithm compares `KEY` values that haven't been initialized. If such behavior is too shocking, it can be avoided by setting all `KEY`s to 0, say, in step R1.

Page 647 new paragraph to insert before answer 27 ———— 24 May 1995

The formula in (b) is due to P. Appell, *Archiv der Math. und Physik* **65** (1880), 171–175. We have, incidentally, $\left[\!\left[{r \atop k} \right]\!\right] = (r+k)!\,[x^k z^r]\,e^{xf(z)}$, where $f(z) = z/2 + z^2/3 + \cdots = -z^{-1}\ln(1-z) - 1$; hence $c_r = [z^r]\,\bigl(r+1+f(z)\bigr)e^{f(z)}$.

Page 648 replacement for lines 1–3 of answer 7 ———— 04 Nov 1996

7. Let $\alpha_p = 2x$ and $z = -1/2^{p+1}$. Then $x^{p+1} = x^p + z$, so we have the convergent series $\alpha_p = 2\sum_{k\ge 0}\binom{1-kp}{k} z^k/(1-kp) = 2 - 2^{-p} - p2^{-2p-1} + O(p^2 2^{-3p})$ by Eq. 1.2.6–21.

Page 651 new answer ———— 05 Nov 1996

31. [*Random Structures & Algorithms* **5** (1994), 102–104.] $K_d(n) = F^{(d)}_{n-2} = N^{(d)}_{n-d-1}$. We have $n-d-1 = a_1 + \cdots + a_r$ if the tree has $r+1$ leaves and the $k+1$st leaf has $a_k - 1$ ancestors distinct from the ancestors of the first k leaves. (The seven example trees correspond respectively to $1+1+1+1$, $1+1+2$, $1+2+1$, $1+3$, $2+1+1$, $2+2$, and $3+1$.)

▸**Page 651** answer 2 line 4 ———— 02 May 1976

D8 $\rightsquigarrow$ C8

▸**Page 658** line 7 ———— 27 May 1978

increase l by 1, set ..., and return $\rightsquigarrow$ set ..., increase l by 1, and return

▸**Page 659** line 3 of answer 8 ———— 04 Feb 1997

miniscule $\rightsquigarrow$ minuscule

Page 664 new answer ———— 14 May 1976

10. See *STOC* **6** (1974), 216–229.

Page 665 new answer ——— 10 Oct 1980

19. There are at least $(nc)!/c!^n$ configurations, and the number that can be obtained from a given one after s stops is at most $\bigl((n-1)\binom{b+c}{b}\bigr)^s$, which is less than $n^s 2^{(b+c)s}$. Hence $s > \bigl(\ln(nc)! - n\ln c!\bigr)/\bigl(\ln n + (b+c)\ln 2\bigr)$ and the stated results follow.

Page 665 answer to 5.5–3 ——— 14 May 1976

line 7: Trabb-Pardo $\leadsto$ Trabb Pardo
line 8: answer $\leadsto$ asymptotic answer
line 10: variables. $\leadsto$ variables, without transforming the records being sorted in any way [*SICOMP* **6** (1977), 351–372]. The same time and space bounds have been achieved with much smaller constant factors by B.-C. Huang and M. A. Langston, *Comp. J.* **35** (1992), 643–650. See also A. Symvonis, *Comp. J.* **38** (1995), 681–690, for cases where m is much smaller than n.

Page 666 replacement for answer 9 ——— 25 May 1996

9. Yes: (a) $\overline{C}_N = N - N^{-\theta}H_{N-1}^{(-\theta)} = \frac{\theta}{1+\theta}N + \frac{1}{2} + O(N^{-\theta})$. (b) $\overline{C}_N = \frac{\theta}{1+\theta}\bigl(1 + N/(1-\binom{N-\theta}{N})\bigr) = \frac{\theta}{1+\theta}\bigl(N + N^{1-\theta}/\Gamma(1-\theta) + 1\bigr) + O(N^{1-2\theta})$.

When $\theta < 0$, (11) is not a probability distribution; (16) gives the estimate $\overline{C}_N = -\frac{\theta}{1+\theta}\Gamma(1-\theta)N^{1+\theta} + O(N^{1+2\theta}) + O(1)$ instead of (15).

Page 666 new paragraph to follow answer 11 ——— 04 Jan 1997

Notes: W. J. Hendricks [*J. Applied Probability* **9** (1972), 231–233] found a simple formula for the steady-state probability of each permutation of the records. For example, when $N = 4$ the sequence will be $R_3\,R_1\,R_4\,R_2$ with limiting probability

$$\frac{p_3}{p_3+p_1+p_4+p_2}\;\frac{p_1}{p_1+p_4+p_2}\;\frac{p_4}{p_4+p_2}\;\frac{p_2}{p_2}.$$

James Bitner [*SICOMP* **8** (1979), 82–85] proved that, if the list is originally in random order, the expected search time after t random requests exceeds $\overline{C}_N$ by the quantity $\frac14\sum_{i,j}(p_i-p_j)^2(1-p_i-p_j)^t/(p_i+p_j)$. Thus, t searches require fewer than $t\overline{C}_N + \frac14\sum_{i,j}(p_i-p_j)^2/(p_i+p_j)^2 < t\overline{C}_N + \frac12\binom{N}{2}$ comparisons altogether, on the average.

Page 667 answer 19 ——— 01 Jun 1981

line 1: We $\leadsto$ Assuming that $d(i,i) = 0$, we
line 3: is due $\leadsto$ for $i \neq j$ is due

▸**Page 667** line 8 of answer 19 ——— 31 Dec 1997

et. al. $\leadsto$ et al.

▸**Page 667** line −6 ——— 14 May 1976

Strauss $\leadsto$ Straus

▸**Page 669** answer 18 ——— 03 Jan 1998

line 2: $C_{k-1} \leadsto C_{k-1}(F_k-1)/(F_{k+1}-1)$
line 3: $C'_{k-1} \leadsto C'_{k-1}F_k/F_{k+1}$

Page 670 new answer ——— 24 Aug 1996

22. *FOCS* **17** (1976), 173–177. See also Y. Perl, A. Itai, and H. Avni, *CACM* **21** (1978), 550–554; G. H. Gonnet, L. D. Rogers, and J. A. George, *Acta Informatica* **13** (1980), 39–52; G. Louchard, *RAIRO Inform. Théor.* **17** (1983), 365–385. The variance is $O(\log\log N)$. Extensive empirical tests by G. Marsaglia and B. Narasimhan, *Computers and Math.* **26**, 8 (1993), 31–42, show that the average number of table accesses is very close to $\lg\lg N$, plus about 0.7 if the search is unsuccessful. When $N = 2^{20}$, for example, a random successful search in a random table takes about 4.29 accesses, while a random unsuccessful search takes about 5.05.

Page 670 new answer —— 12 Nov 1991

28. [*Skrifter udgivne af Videnskabs-Selskabet i Christiania*, Mathematisk-Naturvidenskabelig Klasse (1910), No. 8; reprinted in Thue's *Selected Mathematical Papers* (Oslo: Universitetsforlaget, 1977), 273–310.] (a) T_n has $F_{n+1}+F_{n-1} = F_{2n}/F_n$ leaves. (This is the so-called Lucas number $L_n = \phi^n + \widehat{\phi}^{\,n}$.) (b) The axiom says that $T_0(T_2(x)) = T_1(x)$, and we obviously have $T_m(T_n(x)) = T_{m+n-1}(x)$ when $m = 1$ or $n = 1$. By induction on n, the result holds when $m = 0$; for example, $T_0(T_3(x)) = T_0(T_2(x) * T_1(x)) = T_0(T_1(T_2(x)) * T_0(T_2(x))) = T_0(T_2(T_2(x))) = T_2(x)$. Finally we can use induction on m.

Page 670 new answers —— 21 Jun 1991

29. Assume that $K_0 = -\infty$ and $K_{N+1} = K_{N+2} = \infty$. First do a binary search on $K_2 < K_4 < \cdots$; this takes at most $\lfloor \lg N \rfloor$ comparisons. If unsuccessful, it determines an interval with $K_{2j-2} < K < K_{2j}$; and K is not present if $2j = N+2$. Otherwise, a binary search for K_{2j-1} will determine i such that $K_{2i-2} < K_{2j-1} < K_{2i}$. Then either $K = K_{2i-1}$ or K is not present.

30. Let $n = \lfloor N/4 \rfloor$. Starting with $K_1 < K_2 < \cdots < K_N$, we can put K_1, K_3, $\ldots$, K_{2n-1} into any desired order by swapping them with a permutation of K_{2n+1}, K_{2n+3}, $\ldots$, K_{4n-1}; this arrangement satisfies the conditions of the previous exercise. Now we let $K_1 < K_3 < \cdots < K_{2^{t+1}-3}$ represent the boundaries between all possible t-bit numbers, and we insert $K_{2^{t+1}-1}$, $K_{2^{t+1}+1}$, $\ldots$, $K_{2^{t+1}+2m-3}$ between these "fenceposts" according to the values of x_1, x_2, $\ldots$, x_m. For example, if $m = 4$, $t = 3$, $x_1 = (001)_2$, $x_2 = (111)_2$, and $x_3 = x_4 = (100)_2$, the desired order is

$$K_1 < K_{15} < K_3 < K_5 < K_7 < K_{19} < K_{21} < K_9 < K_{11} < K_{13} < K_{17}.$$

(We could also let K_{21} precede K_{19}.) A binary search for $K_{2^{t+1}+2j-3}$ in the subarray $K_1 < K_3 < \cdots < K_{2^{t+1}-3}$ will now find the bits of x_j from left to right. [Cf. Fiat, Munro, Naor, Schäffer, Schmidt, and Siegel, *J. Comp. Syst. Sci.* **43** (1991), 406–424.]

▶**Page 671** answer 2 —— 28 Feb 1978

line 1: `RTAG` $\rightsquigarrow$ `RTAG(Q)`
line 2: `RLINK(P).` $\rightsquigarrow$ `RLINK(P)` and `RTAG(P)` $\leftarrow +$. In step T4, change the test "`RLINK(P)` $\neq \Lambda$" to "`RTAG(P)` $\neq +$".
last line: .] $\rightsquigarrow$. Similar remarks apply with simultaneous left and right threading.]

▶**Page 671** line 1 of answer 4 —— 11 Jan 1998

$C'_{2^n} \rightsquigarrow C'_{2^n-1}$

Page 672 answer 7 —— 01 Jan 1985

Set up ... $C_{nm} = H_m + H_{n+1-m} - 1$. $\rightsquigarrow$ A comparison with the kth largest element will be made if and only if that element occurs before the mth and before all those between the kth and mth; this happens with probability $1/(|m-k|+1)$. Summing over k gives the answer $H_m + H_{n+1-m} - 1$.

Page 672 end of answer 7 ——— 14 May 1976

77–80]. $\leadsto$ 77–80; see also L. Guibas, *Acta Informatica* **4** (1975), 293–298.]

▶**Page 672** line 1 of answer 15 ——— 23 Oct 1979

$a_i \leadsto a_j$

▶**Page 673** tree illustration in answer 23 ——— 15 Nov 1978

$5 \leadsto 9$

▶**Page 673** answer 27 starting at the displayed paths ——— 19 Mar 1975

$$r_0,\ r_1,\ \ldots,\ \boxed{r_l} \qquad \text{and} \qquad s_0,\ s_1,\ \ldots,\ \boxed{s_{l'}}.$$

Since $r = r_0 > s_0 = s$ and $r_{l'} < s_{l'} = n$, we can find a level $k \geq 0$ such ...
the right subtrees of r_k and s_k can be interchanged, and $\leadsto$

Page 674 new answer ——— 14 May 1976

30. See R. L. Wessner, *Information Processing Letters* **4** (1976), 90–94.

Page 674 new answer ——— 26 Oct 1996

33. For the stated problem, see A. Itai, *SICOMP* **5** (1976), 9–18. For the alternatives, see D. Spuler, *Acta Informatica* **31** (1994), 729–740.

Page 674 replacement for answer 36 ——— 13 Feb 1995

36. See Altenkamp and Mehlhorn, *JACM* **27** (1980), 412–427, for generalizations to multiway trees.

Page 674 new answer ——— 28 Dec 1996

41. Don't let the complicated analyses of the cases $N = 3$ [Jonassen and Knuth, *J. Comp. Syst. Sci.* **16** (1978), 301–322] or $N = 4$ [Baeza-Yates, *BIT* **29** (1989), 378–394] scare you; think big!

Page 674 new answers ——— 13 Jul 1996

42. **U1.** [Initialize.] Set $\texttt{P} \leftarrow \texttt{ROOT}$, $\texttt{R} \leftarrow \Lambda$.

U2. [Compare.] If $K \geq \texttt{KEY(P)}$, go to U4.

U3. [Move left.] If $\texttt{LLINK(P)} \neq \Lambda$, set $\texttt{P} \leftarrow \texttt{LLINK(P)}$ and go back to U2. Otherwise go to U5.

U4. [Move right.] Set $\texttt{R} \leftarrow \texttt{P}$. Then if $\texttt{RLINK(P)} \neq \Lambda$, set $\texttt{P} \leftarrow \texttt{RLINK(P)}$ and go back to U2.

U5. [Success?] If $\texttt{R} \neq \Lambda$ and $K = \texttt{KEY(R)}$, the search terminates successfully.

U6. [Insert into tree.] Same as step T5 of Algorithm T. ▮

[A. Andersson, *Software Practice & Experience* **21** (1991), 1125–1128.]

43. The analysis of Algorithm T applies to Algorithm U without change with respect to unsuccessful searches, since both Algorithms run through the same paces. But in a random successful search, Algorithm U reaches each of the $N+1$ external nodes except the leftmost with probability $1/N$. Therefore step U2 is performed $\bigl((N+1)C'_N - H_N\bigr)/N = C_N + 1 - H_N/N$ times, on the average, where C'_N and C_N are given by (5) and (6). Step U4 is performed after $\frac{1}{2}C_N + \frac{1}{2}$ of those comparisons.

Page 674 new answer ——— 09 Sep 1996

44. This question was first investigated by J. M. Robson [*Australian Comp. J.* **11** (1979), 151–153], B. Pittel [*J. Math. Anal. Applic.* **103** (1984), 461–480], and Luc Devroye [*JACM* **33** (1986), 489–498; *Acta Inf.* **24** (1987), 277–298], who obtained limit formulas that hold with probability $\to 1$ as $n \to \infty$; see the exposition by H. M. Mahmoud, *Evolution of Random Search Trees* (Wiley, 1992), Chapter 2. Sharper results were subsequently found by Luc Devroye and Bruce Reed, *SICOMP* **24** (1995), 1157–1162, who proved that the average height is $\alpha \ln n + O(\log\log n)$ and the variance is $O(\log\log n)^2$, where

$$\alpha = 1/T(1/2e) \approx 4.31107\,04070\,01005\,03504\,70760\,96446\,89027\,83916-$$

and $T(z) = \sum_{n=1}^{\infty} n^{n-1}z^n/n!$.

▸**Page 674** answer 2 ——— 16 Jan 1998

$\texttt{B(S)} = \text{0} \rightsquigarrow \texttt{B(S)} = 0$

Page 675 new paragraph for the end of answer 4 ——— 07 Oct 1996

The Fibonacci tree of order n does, however, minimize the value of $(n+a)N - \bigl(\text{external path length}(T)\bigr)$ over all balanced trees T of height $n-1$, when a is any nonnegative constant; this is readily proved by induction on n. Its external path length is $\frac{3}{5}nF_{n-1} + \frac{4}{5}(n-1)F_n = (\phi/\sqrt{5})nF_{n+1} + O(F_{n+1}) = \Theta(n\phi^n)$. Consequently the path length of any N-node balanced tree is at most

$$\min_n\bigl(nN - \Theta(n\phi^n) + O(N)\bigr) \le N\log_\phi N - N\log_\phi\log_\phi N + O(N).$$

Moreover, if N is large and $k = \lceil \lg N\rceil$, $n = \lfloor k/\lg\phi - \log_\phi k\rfloor = \log_\phi N - \log_\phi\log_\phi N + O(1)$, we can construct a balanced tree of path length $nN + O(N)$ as follows: Write $N + 1 = F_n + F_{n-1} + \cdots + F_{k+1} + N' = F_{n+2} - F_{k+2} + N'$, and construct a complete binary tree on N' nodes; then successively join it with Fibonacci trees of orders k, $k+1$, $\ldots$, $n-1$. [Cf. R. Klein and D. Wood, *Theoretical Comp. Sci.* **72** (1990), 251–264.]

▸**Page 675** line 4 of answer 7 ——— 16 Jan 1998

$\lfloor\theta^{2n}\rfloor \rightsquigarrow \lfloor\theta^{2^n}\rfloor$

▸**Page 675** lines 2 and 3 of answer 8 ——— 16 Jan 1998

$b_h = 2b_{h-1} - \epsilon_h$ ... extremely small $\rightsquigarrow$ $b_{h+1} = 2b_h - \epsilon_h$; hence $b_h = 2^h(1 - \epsilon_1/4 - \epsilon_2/8 - \cdots) + r_h$, where $r_h = \epsilon_h/2 + \epsilon_{h+1}/4 + \cdots$ is extremely small

Page 675 new answer ——— 12 Apr 1995

9. Andrew Odlyzko has shown that the number of balanced trees is asymptotically $c^n f(\log_{(\sqrt{10}+2)/3} n)/n$, where $c \approx 1.916067$ and $f(x) = f(x+1)$. [See *Congressus Numerantium* **42** (1984), 27–52, a paper in which he also discusses the enumeration of 2-3 trees.]

Page 675 new answer ——— 18 Feb 1995

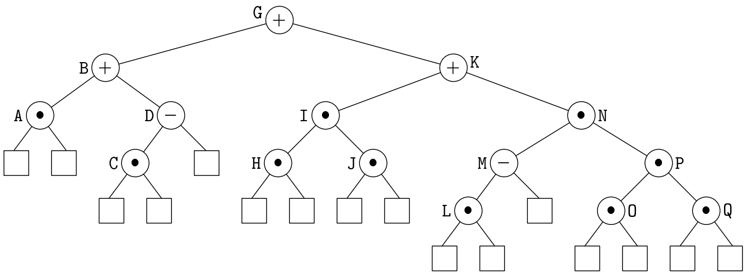
10. [*Inf. Proc. Letters* **17** (1983), 17–20.] To construct the tree, set $\texttt{P} \leftarrow 0$ and compute $\texttt{TREE}(\infty)$, where $\texttt{TREE}(hmax)$ is the following recursive procedure with local variables h, h', and $\texttt{Q}$: Set $h \leftarrow 0$, $\texttt{Q} \leftarrow \Lambda$; then while $h < hmax$ and $\texttt{P} < N$ set $\texttt{P} \leftarrow \texttt{P} + 1$, $h' \leftarrow h + \texttt{B(P)}$, $\texttt{LEFT(P)} \leftarrow \texttt{Q}$, $\texttt{RIGHT(P)} \leftarrow \texttt{TREE}(h')$, $h \leftarrow \max(h, h') + 1$, $\texttt{Q} \leftarrow \texttt{P}$; return $\texttt{Q}$. (Tree $\texttt{Q}$ has height h and corresponds to the balance factors that have been read since entry to the procedure.) The algorithm works even if $|\texttt{B(P)}| > 1$.

Page 675 new answer ——— 29 Nov 1977

11. Clearly there are as many `+A`'s as `--B`'s and `+-B`'s, when $n \geq 2$, and there is symmetry between `+` and `-`. If there are M nodes of types `+A` or `-A`, consideration of all possible cases when $n \geq 1$ shows that the next random insertion results in $M-1$ such nodes with probability $3M/(n+1)$, otherwise it results in $M+1$ such nodes. The result follows. [*SICOMP* **8** (1979), 33–41; Kurt Mehlhorn extended the analysis to deletions in *SICOMP* **11** (1982), 748–780. See R. A. Baeza-Yates, *Computing Surveys* **27** (1995), 109–119, for a summary of later developments in "fringe analyses" such as these.]

Page 676 replacement for answer 16 ——— 29 Nov 1977

16. Delete `E`; do Case 3 rebalancing at `D`. Delete `G`; replace `F` by `G`; do Case 2 rebalancing at `H`; adjust balance factor at `K`.



▸**Page 676** answer 19 ——— 14 May 1976

The fourth rectangle in the left-hand figure is too short — it should be extended so that its bottom line is at the same level as the bottom of the first and third rectangles.

Page 677 replacement for last line of answer 20 ——— 30 May 1995

Although is difficult to insert a new node at the extreme left of the tree shown here, K.-J. Räihä and S. H. Zweben have devised a general insertion algorithm that takes $O(\log N)$ steps. [*CACM* **22** (1979), 508–512.]

Page 678 new answer ——— 14 May 1976

29. Partial solution by A. Yao: With $N \geq 6$ keys the lowest level will contain an average of $\frac{2}{7}(N+1)$ one-key nodes and $\frac{1}{7}(N+1)$ two-key nodes. The average total number of nodes lies between $0.70N$ and $0.79N$, for large N. [*Acta Informatica* **9** (1978), 159–170.]

▸**Page 678** answer 30 line 4 ——— 05 Jun 1975

left subtree of that node $\rightsquigarrow$ subtree rooted at that node

Page 678 new paragraph for the end of answer 30 ——— 28 Jul 1996

An improved method has been developed by R. P. Brent, *ACM Trans. Prog. Languages and Systems* **11** (1989), 388–403.

Page 678 new answer ——— 14 May 1976

31. Use a nearly balanced tree, with `RANK` fields and with additional upward links for the leftmost part, plus a stack of postponed balance factor adjustments along this path. Each insertion does a bounded number of these adjustments per node visited in the path. [Stanford Comp. Sci. report 606 (1977), 60–62. Many other solutions to this problem have been discovered subsequently; see, for example, Brown and Tarjan, *SICOMP* **9** (1980), 594–614.]

Page 679 changes to answer 5 ———— 14 Jun 1977

5. 450. The worst ... chars. $\leadsto$

5. Interpretation 1, trying to maximize the stated minimum: 450. (The worst ... chars.)

Interpretation 2, trying to equalize the number of keys after splitting, in order to keep branching factors high: 155 (15 short keys followed by 16 long ones).

See E. M. McCreight, *CACM* **20** (1977), 670–674, for further comments.

Page 679 addendum to answer 7 ———— 23 May 1996

With variable-length keys as in exercise 5, a parent node may need to split when one of its keys becomes longer.

▸**Page 680** near the top ———— 18 Jan 1998

line 10: $((1 \leadsto (1$
line 11: $\sum_{1\le j\le M} \leadsto \sum_{1\le i\le M}$

Page 680 bottom, new paragraphs for answer 4 ———— 25 Mar 1981

A more versatile way to economize on trie storage has been proposed by Kurt Maly, *CACM* **19** (1976), 409–415.

In general, if we want to compress n sparse tables containing respectively x_1, $\ldots$, x_n nonzero entries, a 'first-fit' method that offsets the jth table by the minimum amount r_j that will not conflict with the previously placed tables will have $r_j \le (x_1 + \cdots + x_{j-1})x_j$, since each previous nonzero entry can block at most x_j offsets. This worst-case estimate gives $r_j \le 93$ for the data in Table 1, guaranteeing that any twelve tables of length 30 containing respectively 10, 5, 4, 3, 3, 3, 3, 3, 2, 2, 2, 2 nonzero entries can be packed into $93 + 30$ consecutive locations regardless of the pattern of the nonzeros. Further refinements of this method have been developed by R. E. Tarjan and A. C. Yao, *CACM* **22** (1979), 606–611.

▸**Page 682** new answer (extends to page 683) ———— 05 Jun 1975

11. No; eliminating a node with only one empty subtree will "forget" one bit in the keys of the nonempty subtree. To delete a node, we should replace it by one of its *terminal* descendants, e.g., by searching to the right whenever possible.

▸**Page 683** answer 12 ———— 14 May 1976

line 3: Algorithm 6.2.2D. $\leadsto$ the algorithm suggested in the previous exercise.
last line: $\frac{3}{8} \leadsto \frac{1}{4}$

▸**Page 684** line -8 ———— 28 Feb 1978

$f_0(n-1) \leadsto f_0(N-1)$

Page 686 new answer ———— 30 May 1995

29. Flajolet and Sedgewick [*SICOMP* **15** (1986), 748–767] have shown that the average number of such nodes is approximately $.372N$ when $M = 2$; $.689N$ when $M = 16$.

▸**Page 686** answer 34 ——— 04 Nov 1975

line 1: $B_k 2^{j(k-1)}$ ⇝ $B_k/2^{j(k-1)}$

line 2: (b) Consider $\bigl(B_n(m) - B_n\bigr)/nm^{n-1}$. ⇝ (b) In the $1/(e^x - 1)$ part, it suffices to consider values of j with $x \le 2\ln n$. For $1 \le x \le 2\ln n$ we have $\sum_{1\le k<n/x}(1 - kx/n)^{n-1} = \sum_{k\ge1} e^{-kx} + O(x^2e^{-x}/n)$. For $x \le 1$ we have $\sum_{k=0}^{n-1}\binom{n}{k}B_k(x/n)^k = \sum_{k\ge0} B_k x^k/k! + O(x^2/n)$.

line 4: $+f(n)$, ⇝ $+f(n) + 2/n$, $\sum_{k\le1}$ ⇝ $\sum_{k\ge1}$

line 5: $\lg(n/\pi)$ ⇝ $\lg(n/\pi))$

Page 687 new answer ——— 14 May 1976

39. See Miyakawa, Yuba, Sugito, and Hoshi, *SICOMP* **6** (1977), 201–234.

▸**Page 687** answer 1 ——— 28 Feb 1978

-38 ⇝ -37

Page 687 replacement for opening lines of answer 4 ——— 14 Jun 1977

4. Consider cases with k pairs. The smallest n such that

$$m^{-n}n!\sum_{0\le k\le n}\binom{m}{n-k}\binom{n-k}{k}2^{-k} < \frac{1}{2}, \qquad \text{for } m = 365,$$

Page 688 last line of answer 8 ——— 22 Aug 1995

formed. ⇝ formed. [Further structure can be deduced by generalizing the Fibonacci number system of exercise 1.2.8–34; see L. H. Ramshaw, *J. Number Theory* **13** (1981), 138–175.]

Page 688 new answer ——— 01 Mar 1979

10. See F. M. Liang's elegant proof in *Discrete Math.* **28** (1979), 325–326.

Page 689 line 2 ——— 16 Mar 1981

lists, ⇝ lists, following a suggestion of Allen Newell,

Page 689 answer 14 ——— 23 Feb 1979

new paragraph to insert at the beginning of the answer:

14. According to the stated conventions, the notation "`X` ⇐ `AVAIL`" of 2.2.3–(6) now stands for the following operations: "Set `X` ← `AVAIL`; then set `X` ← `LINK(X)` zero or more times until either `X` = Λ (an `OVERFLOW` error) or `TAG(X)` = 0; finally set `AVAIL` ← `LINK(X)`."

line 2: keys ⇝ all keys

replace lines 12–13 by: until either `P` = Λ, or `TAG(P)` = 1 and either `AUX(P)` = K [if all keys are short] or `AUX(P)` points to a word containing K (perhaps indirectly through words with `TAG` = 2).

new paragraph to insert at the end of the answer:

Another way to place a hash table "on top of" a large linked memory, using coalescing lists instead of separate chaining, has been suggested by J. S. Vitter [*Inf. Proc. Letters* **13** (1981), 77–79].

Page 690 replacement for answer 23 ——— 06 Jun 1980

23. A deletion scheme for coalesced hashing devised by J. S. Vitter [*J. Algorithms* **3** (1982), 261–275] preserves the distribution of search times.

Page 691 line 12 ——— 14 May 1976

and $\rightsquigarrow$ with $0/0 = 1$ when $k = N = M - 1$, and

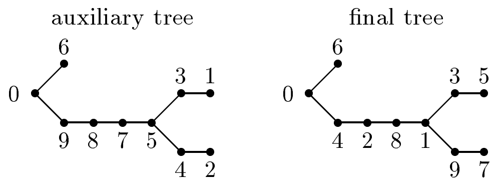
Page 691 end of answer 29 ——— 26 Jun 1996

1274, who were … . $\rightsquigarrow$ 1274.

Page 692 answer 31 ——— 19 Jun 1996

line 1: two $\rightsquigarrow$ three
line 2: favorites: $\rightsquigarrow$ favorites [see also Foata and Riordan, *Æquat. Math.* **10** (1974), 10–22]:
replacement for final paragraph:

c) First construct an auxiliary tree by letting the parent of node k be the first element $> k$ that follows k in the permutation $q_1 \ldots q_n$; if there's no such element, let the parent be 0. Then make a copy of the auxiliary tree and relabel the nonzero nodes of the new tree by proceeding as follows, in preorder: If the label of the current node was k in the auxiliary tree, swap its current label with the label that is currently $(1 + p_k - a_k)$th smallest in its subtree. For example,



To reverse the procedure, we can reconstruct the auxiliary tree by proceeding in preorder to swap the label of each node with the largest label currently in its subtree.

Constructions (a) and (b) are strongly related, but construction (c) is quite different. It has the interesting property that the sum of displacements of cars from their preferred locations is equal to the number of *inversions* in the tree — the number of pairs of labels $a > b$ where a is an ancestor of b. This relation between parking sequences and tree inversions was first discovered by G. Kreweras [*Periodica Math. Hung.* **11** (1980), 309–320]. The fact that tree inversions are intimately related to connected graphs [Mallows and Riordan, *Bull. Amer. Math. Soc.* **74** (1968), 92–94] now makes it possible to deduce that the sum of $\binom{D(p)}{k}$ taken over all parking sequences, where $D(p) = (p_1 - a_1) + \cdots + (p_n - a_n)$, is equal to the total number of connected graphs with $n + k$ edges on the labeled vertices $\{0, 1, \ldots, n\}$. [See equations (2.11), (3.5), and (8.13) in the paper by Janson, Łuczak, Knuth, and Pittel, *Random Struct. and Alg.* **4** (1993), 233–358.]

Page 692 bottom two lines ——— 01 Jun 1995

[A slightly … (to appear).] $\rightsquigarrow$

Page 693 new paragraph to follow answer 33 ——— 24 Aug 1996

Gaston Gonnet and Ian Munro [*J. Algorithms* **5** (1984), 451–470] have found an interesting way to derive the exact result from the argument leading up to (51) by introducing a useful operation called the *Poisson transform* of a sequence $\langle A_{mn} \rangle$: We have $e^{-mz} \sum_n A_{mn}(mz)^n/n! = \sum_k a_k z^k$ if and only if $A_{mn} = \sum_k a_k n^{\underline{k}}/m^k$.

Page 693 answer 34 ______ 10 Jan 1980

lines 4 and 5: C'_N ... all keys. $\rightsquigarrow$ Consider the total number of probes to find all keys, not counting the fetching of the pointer in the list head table of Fig. 38 if such a table is used.

line −1: Thus we obtain (18), (19). $\rightsquigarrow$ (d) In case (i) a list of length k requires k probes (not counting the list-head fetch), while in case (ii) it requires $k+\delta_{k0}$. Thus in case (ii) we get $C'_N = \sum(k+\delta_{k0})P_{Nk} = P'_N(1) + P_N(0) = N/M + (1 - 1/M)^N \approx \alpha + e^{-\alpha}$, while case (i) has simply $C'_N = N/M = \alpha$. The formula $MC'_N = M - N + NC_N$ applies in case (iii), since $M-N$ hash addresses will discover an empty table position while N will cause searching to the end of some list; this yields (18).

Page 693 replacement for answer 35 ______ 10 Jan 1980

35. (i) $\sum\bigl(1+\frac12 k-(k+1)^{-1}\bigr)P_{Nk} = 1 + N/(2M) - M\bigl(1-(1-1/M)^{N+1}\bigr)/(N+1) \approx 1+\frac12\alpha-(1-e^{-\alpha})/\alpha$. (ii) Add $\sum\delta_{k0}P_{Nk} = (1-1/M)^N \approx e^{-\alpha}$ to the result of (i). (iii) Assume that when an unsuccessful search begins at the jth element of a list of length k, the given key has random order with respect to the other k elements, so the expected length of search is $\bigl(j\cdot 1 + 2 + \cdots + (k+1-j) + (k+1-j)\bigr)/(k+1)$. Summing on j now gives $MC'_N = M - N + M\sum(k^3+9k^2+2k)P_{Nk}/(6k+6) = M - N + M\bigl(\frac16 N(N-1)/M^2 + \frac32 N/M - 1 + \bigl(M/(N+1)\bigr)\bigl(1-(1-1/M)^{N+1}\bigr)\bigr)$; hence $C'_N \approx \frac12\alpha + \frac16\alpha^2 + (1-e^{-\alpha})/\alpha$.

Page 693 answer 36 ______ 06 Jun 1980

line 1, replace first sentence by: (i) $N/M - N/M^2$. (ii) $\sum(\delta_{k0}+k)^2P_{Nk} = \sum(\delta_{k0}+k^2)P_{Nk} = P_N(0) + P''_N(1) + P'_N(1)$.

line −1, add new remark: [For data structure (iii), a more complicated analysis like that in exercise 37 would be necessary.]

▸**Page 693** replacement for all but the first line of answer 37 ______ 19 Dec 1976

$$\begin{aligned}
M^N N S_N &= \frac13\sum\binom{N}{k_1,\ldots,k_M}\bigl(k_1(k_1-\tfrac12)(k_1-1)+\cdots+k_M(k_M-\tfrac12)(k_M-1)\bigr)\\
&= \frac13 M\sum_k\binom{N}{k}(M-1)^{N-k}k(k-\tfrac12)(k-1)\\
&= \frac13 MN(N-1)(N-2)\sum_k\binom{N-3}{k-3}(M-1)^{N-k}\\
&\qquad + \frac12 MN(N-1)\sum_k\binom{N-2}{k-2}(M-1)^{N-k}\\
&= \frac13 MN(N-1)(N-2)M^{N-3} + \frac12 MN(N-1)M^{N-2}.
\end{aligned}$$

The variance is $S_N - \bigl((N-1)/2M\bigr)^2 = (N-1)(N+6M-5)/12M^2 \approx \frac12\alpha + \frac1{12}\alpha^2$.

See *CMath* §8.5 for interesting connections between the variance calculated here and two other notions of variance: the variance (over random hash tables) of the average number of probes (over all items present), and the average (over random hash tables) of the variance of the number of probes (over all items present).

Page 694 replacement for lines 1–8 of answer 39 ________________ 01 Jun 1980

39. (This approach to the analysis of Algorithm C was suggested by J. S. Vitter.) We have $c_{N+1}(k) = (M-k)c_N(k) + (k-1)c_N(k-1)$ for $k \ge 2$, and furthermore $\sum kc_N(k) = NM^N$. Hence $S_{N+1} = \sum_{k\ge 2}\binom{k}{2}c_{N+1}(k) = \sum_{k\ge 2}\binom{k}{2}\bigl((M-k)c_N(k) + (k-1)c_N(k-1)\bigr) = \sum_{k\ge 1}\bigl((M+2)\binom{k}{2} + k\bigr)c_N(k) = (M+2)S_N + NM^N$.

Page 694 line 1 of answer 40 ________________ 01 Jun 1980

$\binom{j}{2}$ replaced by $\binom{j+1}{3}$ $\leadsto$ $\binom{k}{2}$ replaced by $\binom{k+1}{3}$

Page 694 new answer ________________ 06 Jun 1980

43. Let $N = \alpha M'$ and $M = \beta M'$, and let $e^{-\lambda} + \lambda = 1/\beta$, $\rho = \alpha/\beta$. Then $C_N \approx 1 + \frac{1}{2}\rho$ and $C'_N \approx \rho + e^{-\rho}$, if $\rho \le \lambda$; $C_N \approx \frac{1}{8\rho}(e^{2\rho-2\lambda} - 1 - 2\rho + 2\lambda)(3 - 2/\beta + 2\lambda) + \frac{1}{4}(\rho + 2\lambda - \lambda^2/\rho)$ and $C'_N \approx 1/\beta + \frac{1}{4}(e^{2\rho-2\lambda} - 1)(3 - 2/\beta + 2\lambda) - \frac{1}{2}(\rho - \lambda)$, if $\rho \ge \lambda$. For $\alpha = 1$ we get the smallest $C_N \approx 1.69$ when $\beta \approx .853$; the smallest $C'_N \approx 1.79$ occurs when $\beta \approx .782$. The setting $\beta = .86$ gives near-optimal search performance for a wide range of α. So it pays to put the first collisions into an area that doesn't conflict with hash addresses, even though a smaller range of hash addresses causes more collisions to occur. These results are due to Jeffrey S. Vitter, *JACM* **30** (1983), 231–258.

Page 695 in future . . . . 31 May 1995

Michael Paterson found better ways to solve many of these problems, and his methods are explained in Chapter 3 of Mathematics for the Analysis of Algorithms by Greene and Knuth (Birkhäuser Boston, 1980). I will include at least a hint of his methods in the next edition.

Page 696 new answer ________________ 14 May 1976

45. Yes. See L. Guibas, *JACM* **25** (1978), 544–555.

Page 697 replacement for answer 50 ________________ 26 May 1996

50. And next that $Q_0(M, N-1) = (M/N)\bigl(Q_0(M,N) - 1\bigr)$. In general, $rQ_r(M,N) = MQ_{r-2}(M,N) - (M-N-r)Q_{r-1}(M,N) = M\bigl(Q_{r-1}(M,N+1) - Q_{r-1}(M,N)\bigr)$; $Q_r(M,N-1) = (M/N)\bigl(Q_r(M,N) - Q_{r-1}(M,N)\bigr)$.

Page 698 new answer ________________ 11 Jun 1996

56. See Blake and Konheim, *JACM* **24** (1977), 591–606. Alfredo Viola and Patricio Poblete [*Algorithmica*, to appear] have shown that

$$\begin{aligned} C_{Mb} &= 1 + \frac{M-1}{2Mb} + \frac{1}{b}\sum_{j\ge 2}\binom{bm-1}{j}m^{-j}\sum_{k\ge 1}\binom{j-2}{bk-1}(-1)^{j+bk-1}k^{j-1} \\ &= \sqrt{\frac{\pi M}{8b}} + \frac{1}{3b} + \frac{1}{b}\sum_{j=1}^{b-1}\frac{1}{\bigl(1 - T(e^{2\pi ij/b-1})\bigr)} + \frac{1}{24}\sqrt{\frac{\pi}{2b^3M}} + O(b^{-2}M^{-1}), \end{aligned}$$

where T is the tree function of Eq. 2.3.4.4–(28).

Page 698 new answer ________________ 11 Nov 1995

60. Preliminary results have been obtained by Ajtai, Komlós, and Szemerédi, *Information Processing Letters* **7** (1978), 270–273.

Page 698 answer 63 ________________ 30 May 1995

[This is the "mean time to failure" ... "deleted."] $\leadsto$

Page 698 new answer ________ 30 May 1995

62. See the discussion in *AMM* **81** (1974), 323–343, where the best cyclic hashing sequences are exhibited for $M \le 9$.

Page 698 new answer ________ 30 May 1995

64. The average number of moves is equal to $\frac{1}{2}(N-1)/M + \frac{2}{3}(N-1)(N-2)/M^2 + \frac{3}{4}(N-1)(N-2)(N-3)/M^3 + \cdots \approx \frac{1}{1-\alpha} - \frac{1}{\alpha}\ln\frac{1}{1-\alpha}$. [An equivalent problem is solved in *Comp. J.* **17** (1974), 139–140.]

Page 699 answer 66 line 3 ________ 30 May 1995

inserted. $\leadsto$ inserted. This important technique is called *ordered hashing*.
142. $\leadsto$ 142; D. Knuth, *Literate Programming* (Stanford, 1992), 144–149, 216–217.

Page 699 new answer ________ 28 May 1996

67. (a) If $c_j = 0$ in (44), an optimum arrangement is obtained by sorting the a's into nonincreasing "cyclic order," assuming that $j-1 > \cdots > 0 > M-1 > \cdots > j$. (b) Between steps L2 and L3, exchange the record-in-hand with `TABLE`$[i]$ if the latter is closer to home than the former. [This algorithm, called "Robin Hood hashing" by Celis, Larson, and Munro in *FOCS* **26** (1985), 281–288, is equivalent to a variant of ordered hashing.] (c) Let $h(m,n,d)$ be the number of hash sequences that make $c_0 \le d$. It can be shown [*Comp. J.* **17** (1974), 141] that $\bigl(h(m,n,d) - h(m,n,d-1)\bigr)M$ is the total number of occurrences of displacement $d > 0$ among all M^N hash sequences, and that we can write $h(M,N,d) = a(M,N,d+1) - Na(M,N-1,d+1)$ where $a(m,n,d) = \sum_{k=0}^{d}\binom{n}{k}(m+d-k)^{n-k}(k-d)^k$. An elaborate calculation using the methods of exercises 28 and 50 now shows that the average value of $\sum d_j^2$ is

$$\begin{aligned} M^{1-N}\sum_{d=1}^{N} d^2\bigl(h(M,N,d) - h(M,N,d-1)\bigr) \\ &= \frac{M^2}{2} + \frac{2M}{3} + \frac{N}{6} + \frac{N^2}{6M} - \frac{N}{6M} - M\Bigl(\frac{M}{2} - \frac{N}{2} + \frac{2}{3}\Bigr)Q_0(M,N) \\ &= M\left(\frac{1}{2(1-\alpha)^2} - \frac{7}{6(1-\alpha)} + \frac{2}{3} + \frac{\alpha}{6} + \frac{\alpha^2}{6}\right) + O(1) \end{aligned}$$

when $N = \alpha M$. Without the modification (see exercise 28), $\sum d_j^2$ comes to

$$\begin{aligned} \frac{M}{3}\bigl(Q_2(M,N) - Q_1(M,N)\bigr) - \frac{M}{2}\bigl(Q_0(M,N) - 1\bigr) + \frac{N}{6} \\ &= M\left(\frac{1}{3(1-\alpha)^3} - \frac{1}{3(1-\alpha)^2} - \frac{1}{2(1-\alpha)} + \frac{1}{2} + \frac{\alpha}{6}\right) + O(1). \end{aligned}$$

If the records all have approximately the same displacement d, and if successful searches are significantly more common than unsuccessful ones, it is advantageous to start at position $h' = h(K) + d$ and then to probe $h'-1$, $h'+1$, $h'-2$, etc. P. V. Poblete, A. Viola, and J. I. Munro have shown [*Random Structures and Algorithms* **10** (1997), 221–255] that $\sum d_j^2$ can be made almost as small as in the Robin Hood method by using a much simpler approach called "last-come-first-served" hashing, in which every newly inserted key is placed in its home position; all other keys move one step away until an empty slot is found. The Robin Hood and last-come-first-served techniques apply to double hashing as well as to linear probing, but the reduction in probes does not compensate for the increased time per probe with respect to double hashing unless the table is extremely full. (See Poblete and Munro, *J. Algorithms* **10** (1989), 228–248.)

Page 699 new answer ——— 14 May 1976

69. Let $q_k = p_k + p_{k+1} + \cdots$; then the inequality $q_k \ge \max\bigl(0, 1-(k-1)(M-n)/M\bigr)$ gives a lower bound on $C'_N = \sum_{k\ge1} q_k$.

Page 699 new answer ——— 08 May 1995

70. A remarkably simple proof by Lueker and Molodowitch [*Combinatorica* **13** (1993), 83–96] establishes a similar result but with an extra factor $(\log M)^2$ in the O bound; the stated result follows in the same way by using sharper probability estimates. A. Siegel and J. P. Schmidt have shown, in fact, that the expected number of probes in double hashing is $1/(1-\alpha)+O(1/M)$ for fixed $\alpha = N/M$. [To appear.]

Page 699 more new answers ——— 30 May 1996

71. [*J. Comp. Syst. Sci.* **18** (1979), 143–154.] (a) Given keys $K_1, \ldots, K_N$ and K, the probability that K_j is in the same list as K is $\le 1/M$ if $K \ne K_j$. Hence the expected list size is $\le 1 + (N-1)/M$.

(b) Suppose there are Q possible characters; then there are M^Q possible choices for each h_j. Choosing each h_j at random is equivalent to choosing a random row from a matrix H of M^{Ql} rows and Q^l columns, with the entry $h(x_1\ldots x_l) = \bigl(h_1(x_1) + \cdots + h_l(x_l)\bigr) \bmod M$ in column $x_1 \ldots x_l$. In columns $K = x_1\ldots x_l$ and $K' = x'_1\ldots x'_l$ with $x_j \ne x'_j$ for some j, we have $h(K) = \bigl(\alpha + h_j(x_j)\bigr) \bmod M$ and $h(K') = \bigl(\alpha' + h_j(x'_j)\bigr) \bmod M$, where $\alpha = \sum_{i\ne j} h_i(x_i)$ and $\alpha' = \sum_{i\ne j} h_i(x'_i)$ are independent of h_j. The value of $h_j(x_j) - h_j(x'_j)$ is uniformly distributed modulo M; hence we have $h(K) = h(K')$ with probability $1/M$, regardless of the values of α and α'.

(c) Yes; adding any constant to $h_j(x_j)$ changes $h(x)$ by a constant, modulo M.

72. (i) This is the special case of exercise 71(c) when each key is regarded as a sequence of bits, not characters. (ii) The proof of (b) shows that it suffices to show that $h_j(x_j) - h_j(x'_j)$ is uniform modulo M when $x_j \ne x'_j$. And in fact, the probability that $h_j(x_j) = y$ and $h_j(x'_j) = y'$ is $1/M^2$, for any given y and y', because the congruences $a_jx_j + b_j \equiv y$ and $a_jx'_j + b_j \equiv y'$ have a unique solution (a_j, b_j) for any given (y, y'), modulo the prime M.

When M is not prime and p is a prime $> M$, a similar result holds if we let $h_j(x_j) = \bigl((a_jx_j + b_j) \bmod p\bigr) \bmod M$, where a_j and b_j are chosen randomly mod p. In this case the family is not quite universal, but it comes close enough for practical purposes: The probability that different keys collide is at most $1/M + r(M-r)/Mp^2 \le 1/M + M/4p^2$, where $r = p \bmod M$.

73. The statement is false in general. For example, suppose $M = N = n^2$, and consider the matrix H with $\binom{N}{n}$ rows, one for every way to put n zeros in different columns; the nonzero entries are $1, 2, \ldots, N-n$ from left to right in each row. This matrix is universal because there are $\binom{N-2}{n-2} = \binom{N}{n}\frac{n}{N}\frac{n-1}{N-1} < \binom{N}{n}\bigl(\frac{n}{N}\bigr)^2 = R/M$ matches in every pair of columns. But the number of zeros in every row is $\sqrt{N} \ne O(1) + O(N/M)$.

Notes: This exercise points out that expected list size is quite different from the expected number of collisions when a new key is inserted. Consider letting $h(x_1\ldots x_l) = h_1(x_1)$, where h_1 is chosen at random. This family of hash functions makes the expected size of every list N/M; yet it is certainly not universal, because a set of N keys that have the same first character x_1 will lead to one list of size N and all other lists empty.

The expected number of collisions will be $N(N-1)/2$, but with a universal hash family this number is at most $N(N-1)/2M$, regardless of the set of keys.

On the other hand we *can* show that the expected size of every list is $O(1)+O(N/\sqrt{M}\,)$ in a universal family. Suppose there are z_h zeros in row h. Then that row contains at least $\binom{z_h}{2}$ pairs of equal elements. The maximum of $\sum_{h=1}^{R} x_h$ subject to $\sum_{h=1}^{R}\binom{z_h}{2} \le \binom{N}{2}R/M$ occurs when each z_h is equal to z where $\binom{z}{2}=\binom{N}{2}/M$, namely

$$z=\frac{1}{2}+\sqrt{\frac{1}{4}+\frac{N(N-1)}{M}}<1+\sqrt{\frac{N(N-1)}{M}}\,.$$

74. (a) Obviously true, even if h_2, ..., h_l are identically zero. (b) True, by the answer to 71(b). (c) True. The result is clear if K, K', and K'' all differ in some character position. Otherwise, say $x_j=x'_j\ne x''_j$ and $x_k\ne x'_k=x''_k$. Then $h_j(x_j)+h_k(x_k)$, $h_j(x_j)+h_k(x'_k)$, and $h_j(x''_j)+h_k(x'_k)$ are independent of each other, uniformly distributed, and independent of the other $l-2$ characters of the keys. (d) False. Consider, for example, the case $M=l=2$ with 1-bit characters. Then all four keys hash to the same location with probability 1/4.

75. Use $h(K)=\bigl(h_0(l)+h_1(x_1)+\cdots+h_l(x_l)\bigr)\bmod M$, where each h_j is chosen as in exercise 72. Generate the random coefficients for h_j (and, if desired, precompute its array of values) when a key of length $\ge j$ occurs for the first time. Since l is unbounded, the matrix H is infinite; but only a finite portion is relevant in any particular run of the program.

Page 699 another new answer ——— 11 Jun 1996

76. Let $p\le 2^{-16}$ be the probability that two 32-bit keys have the same image under H. The worst case occurs when two given keys agree in seven of their eight 32-bit subkeys; then the probability of collision is $1-(1-p)^4<4p$. [Cf. Wegman and Carter, *J. Comp. Syst. Sci.* **22** (1981), 265–279.]

Page 699 last line of answer 1 ——— 01 Jun 1995

involve B_i. $\rightsquigarrow$ involve B_i. Further comments appear in a paper by C. Greene and D. J. Kleitman, *J. Combinatorial Theory* **A20** (1976), 80–88.

Page 699 line −17 ——— 27 Sep 1996

N. G. de Bruijn et al. $\rightsquigarrow$ N. G. de Bruijn, C. van Ebbenhorst Tengbergen, and D. Kruyswijk

▸**Page 700** line 1 ——— 01 Jul 1975

$\sum_t p_tP_t$. $\rightsquigarrow$ $\sum_t p_tP_t$, minus the probability that a particular record is a "true drop," namely $\binom{N-q}{r-q}/\binom{N}{r}$, where $N=\binom{n}{k}$.

▸**Page 700** line 10 ——— 24 Jun 1996

$f(n,k,q)$ $\rightsquigarrow$ $f(n,k,q))$

Page 700 new answer ——— 17 Feb 1997

8. (a) When $S=S_00\cup S_11$, we have $f_t(S)=f_t(S_0\cup S_1)+f_{t-1}(S_0)+f_{t-1}(S_1)$. Therefore $f_t(s,m)$ is the minimum of $f_t(s_0,m-1)+f_{t-1}(s_0,m-1)+f_{t-1}(s_1,m-1)$ over all s_0 and s_1 such that $2^{m-1}\ge s_0\ge s_1\ge 0$ and $s_0+s_1=s$. To prove that the minimum occurs for $s_0=\lceil s/2\rceil$ and $s_1=\lfloor s/2\rfloor$, we can use induction on m, the result being clear for $m=1$ Given $m\ge 2$, let $g_t(s)=f_t(s,m-1)$ and $h_t(s)=f_t(s,m-2)$. Then, by induction, $g_t(s_0)+g_{t-1}(s_0)+g_{t-1}(s_1)=h_t(\lceil s_0/2\rceil)+h_{t-1}(\lceil s_0/2\rceil)+h_{t-1}(\lfloor s_0/2\rfloor)+$

$h_{t-1}(\lceil s_0/2\rceil)+h_{t-2}(\lceil s_0/2\rceil)+h_{t-2}(\lfloor s_0/2\rfloor)+h_{t-1}(\lceil s_1/2\rceil)+h_{t-2}(\lceil s_1/2\rceil)+h_{t-2}(\lfloor s_1/2\rfloor)$, which is $\geq g_t(\lceil s_0/2\rceil+\lceil s_1/2\rceil)+g_{t-1}(\lceil s_0/2\rceil+\lceil s_1/2\rceil)+g_{t-1}(\lfloor s_0/2\rfloor+\lfloor s_1/2\rfloor)$. And if $s_0 > s_1+1$, we have $\lceil s_0/2\rceil+\lceil s_1/2\rceil < s_0$, except in the case $s_0 = 2k+1$ and $s_1 = 2k-1$. In the latter case, however, $g_t(s_0)+g_{t-1}(s_0)+g_{t-1}(s_1) \geq h_t(2k+1)+2h_{t-1}(2k) \geq h_t(2k)+2h_{t-1}(2k)$.

(b) Observe that the set S containing the numbers 0, 1, …, $s-1$ in binary notation has the property that $S_0 \cup S_1 = S_0$, and S_0 contains $\lceil s_0/2\rceil$ elements. It follows, incidentally, that $f_t(2^{m-r}, m) = [z^t]\,(1+z)^r(1+2z)^{m-r}$.

Page 700 new paragraph for end of answer 10 ———— 19 Jun 1996

Incidentally, "Steiner triple systems" should not have been named after Steiner, although that name has become deeply entrenched in the literature. Steiner's publication [*Crelle* **45** (1853), 181–182] came several years after Kirkman's, and Felix Klein has noted [*Vorlesungen über die Entwicklung der Math. im 19. Jahrhundert* **1** (Springer, 1926), 128] that Steiner quoted English authors without giving them credit, during the later years of his life. Moreover, the concept had appeared already in two well-known books of J. Plücker [*System der analytischen Geometrie* (1835), 283–284; *Theorie der algebraischen Curven* (1839), 245–247].

Page 700 new answer ———— 02 Mar 1977

16. Let each triple correspond to a codeword, where each codeword has exactly three 1 bits, identifying the elements of the corresponding triple. If u, v, w are distinct codewords, u has at most two 1 bits in common with the superposition of v and w, since it had at most one in common with v or w alone. [Similarly, from quadruple systems of order v we can construct $v(v-1)/12$ codewords, none of which is contained in the superposition of any three others, etc.]

Page 700 new answer ———— 03 Jan 1997

17. (a) There are $2^n(m-n)$ $*$'s altogether, hence $2^n n$ digits, with $2^n n/m$ digits in each column. Half of the digits in each column must be 0. Hence $2^{n-1}n/m$ is an integer, and each column contains $(2^{n-1}n/m)^2$ mismatches. Since each pair of rows has at least one mismatch, we must have $2^n(2^n-1)/2 \leq (2^{n-1}n/m)^2 m$.

(b) Consider the 2^n m-bit numbers that are 0 in $m-n$ specified columns. Half of these have odd parity. A row with $*$ in any of the unspecified columns covers as many evens as odds.

(c) $*000$, $*111$, $0{*}10$, $1{*}10$, $00{*}1$, $10{*}1$, $010*$, $110*$. This one isn't as uniform as (7), because a query like `*01*` hits four rows while `*10*` hits only two. Notice that (7) has cyclic symmetry.

(d) Generate 4^3 rows from each row of (7) by replacing each $*$ by $****$, each 0 by any one of the first four rows, and each 1 by any one of the last four rows. (A similar construction makes an $\mathrm{ABD}(mm', nn')$ from any $\mathrm{ABD}(m,n)$ and $\mathrm{ABD}(m',n')$.)

(e) Given an $\mathrm{ABD}(16,9)$, we can encircle one $*$ in each row in such a way that there are equally many circles in each column. Then we can split each row into two rows, with the circled element replaced by 0 and 1. To show that such encirclement is possible, note that the asterisks of each column can be arbitrarily divided into 32 groups of 7 each; then the 512 rows each contain asterisks of 7 different groups, and the $32 \times 8 = 512$ groups each appear in 7 different rows. Theorem 7.5.1E now guarantees the existence of a perfect matching with exactly one circled element in each row and each group.

References: R. L. Rivest, *SICOMP* **5** (1976), 19–50; A. E. Brouwer, *Combinatorics*, ed. by Hajnal and Sós, *Colloq. Math. Soc. János Bolyai* **18** (1978), 173–184. Brouwer went on to prove that an ABD$(2n, n)$ exists for all $n \geq 32$. The method of part (d) also yields on ABD$(32, 15)$ when (7) is combined with $(*)$.

Page 700 new answers — 17 Feb 1997

18. By exercise 8, the average number with $8-k$ specified bits is $2^{k-3} f_{8-k}(8,8)/\binom{8}{k}$, which has the respective values $(32, 22, \frac{104}{7}, \frac{69}{7}, \frac{45}{7}, \frac{33}{8}, \frac{73}{28}, \frac{13}{8}, 1) \approx (32, 22, 14.9, 9.9, 6.4, 4.1, 2.6, 1.6, 1)$ for $8 \geq k \geq 0$. These are only slightly higher than the values of $32^{k/8} \approx (32, 20.7, 13.5, 8.7, 5.7, 3.7, 2.4, 1.5, 1)$. The worst-case numbers are $(32, 22, 18, 15, 11, 8, 4, 2, 1)$.

19. J. A. La Poutré [*Disc. Math.* **58** (1986), 205–208] showed that an ABD(m, n) cannot exist when $m > \binom{n}{2}$ and $n > 3$; therefore no ABD$(16, 6)$ exists. La Poutré and van Lint [*Util. Math.* **31** (1987), 219–225] proved that there is no ABD$(10, 5)$. We get an ABD$(8, 6)$ from an ABD$(8, 5)$ or ABD$(4, 3)$ using the methods of exercise 17; this produces several nonisomorphic solutions, and additional examples of ABD$(8, 6)$ might also exist. The only remaining possibilities (besides the trivial ABD$(5, 5)$ and ABD$(6, 6)$) are ABD$(8, 5)$ distinct from $(*)$, and perhaps one or more ABD$(12, 6)$.

▸**Page 702** line -20 (the entry for 1 degree of arc) — 19 Jun 1975

1154 ⇝ *1155*

Page 703 replacement for lines 1–7 — 14 May 1976

A few interesting constants without common names have arisen in connection with the analyses of sorting and searching algorithms. These constants have been evaluated to 40 decimal places in Eq. 5.2.3–(19) and in the answers to exercises 5.2.4–13, 6.2.2–44, 6.3–26, and 6.3–27.

Page 704 lines 7–10 — 01 Nov 1995

$$\begin{aligned}
H_{1/5} &= 5 - \tfrac{1}{2}\pi\phi^{3/2}5^{-1/4} - \tfrac{5}{4}\ln 5 - \tfrac{1}{2}\sqrt{5}\ln\phi,\\
H_{2/5} &= \tfrac{5}{2} - \tfrac{1}{2}\pi\phi^{-3/2}5^{-1/4} - \tfrac{5}{4}\ln 5 + \tfrac{1}{2}\sqrt{5}\ln\phi,\\
H_{3/5} &= \tfrac{5}{3} + \tfrac{1}{2}\pi\phi^{-3/2}5^{-1/4} - \tfrac{5}{4}\ln 5 + \tfrac{1}{2}\sqrt{5}\ln\phi,\\
H_{4/5} &= \tfrac{5}{4} + \tfrac{1}{2}\pi\phi^{3/2}5^{-1/4} - \tfrac{5}{4}\ln 5 - \tfrac{1}{2}\sqrt{5}\ln\phi,
\end{aligned}$$

Page 705 and the next several pages — 14 May 1976

see the corrections to pages 607–611 of Volume 1

Page 707 line -8 — 25 Mar 1995

$\|M\|$ ⇝ $|M|$

Page 709 after the entry for var(g) — 03 Aug 1995

$\Pr(S(X))$	probability that statement $S(X)$ is true, for random values of X	1.2.10
$\mathrm{E}\,X$	expected value of X: $\sum_x x\Pr(X = x)$	1.2.10

Page 710 and following ——— 03 Jun 1995

Miscellaneous changes to the existing index of Volume 3 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 723 bottom two lines ——— 01 Jun 1995

For additional definitions . . . 1966).

Page 726 and 727 (the endpapers) ——— 01 Jan 1980

replace by new table from endpapers of Volume 2