Theory and Practice and Fun



Too young to read books himself, Donald Knuth, 4, of 2961 N. 18th st. is nevertheless the youngest member of the public library's Ancient Order of Book Worms. His father reads storybooks to him. Wednesday he gave oral reports on his "reading" in the young people's room of the library. Some of the stories he reported on were "Country Bunny," "Pokey Bear," "When the Root Children Wake Up" and "Babar the King."

KNUTH, DONALD PRE-SCHOOL
2961-A N. 18

1. When the root children wake up
2. Story of Noah's ark
3. Little buffalo boy
4. At the seashore
5. Seraphine went walking
6. Country bunny
7. Babar the king
8. Travels of Babar
9. Pokey Bear
10. Zephir's holidays
11. Nicodemus and the little black pig
12. Little wooden farmer

Ye Ancient Order of Book Worms

Milwaukee Public Library



Know all men by these presents:

Donald Knuth

has been duly made a REGULAR MEMBER of this illustrious society and from this day of August 29, 1942 is entitled to the honor and privileges accorded every genuine lover of books.

E. E. Seelbach
Exalted Corrodentium

Richard Krug
Lord High Chelifer



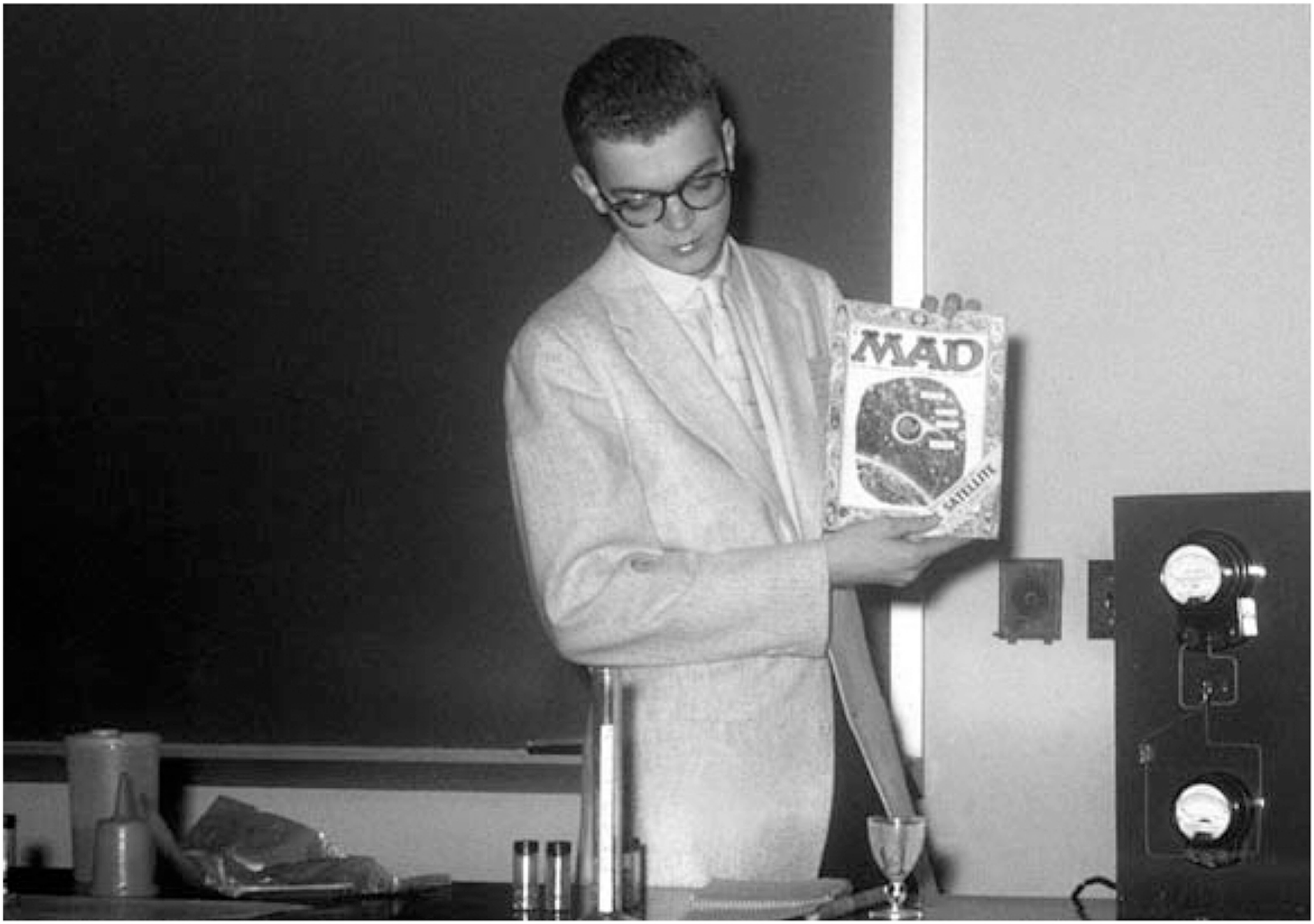


For the record, my list contained just three words of twelve letters or more: `arterializes`, `interregalia`, and the grand champion, `interregalias`. The final entries of my list, alphabetically speaking, were `ziraleet`, `ziraleets`, `zita`, and `zitas`; nowadays Random House would allow me to follow those by `ziti` and `zits`.

At last the big day came, 18 December 1951. Surprise: I'd won!



FIGURE 3. My debut on (low-definition) television, vintage 1951.



Official presentation of the system on 5 May 1956
(photo by Ervin Knuth)

The Wisconsin Junior Academy of Science

Certificate of Merit

to

DONALD KNUTH

In recognition of excellent work in science presented at the STATEWIDE meeting of the Wisconsin Junior Academy of Science during 1956.

S. F. Darling
President, Wisconsin Academy of Sciences, Arts and Letters

John W. Thomson
Chairman, Junior Academy Committee

MAD

225 LAFAYETTE STREET • NEW YORK 12, NEW YORK • CANAL 6-1994

31 July, 1956

Mr. Don Knuth
7436 W. Caldwell Avenue
Milwaukee 16,
Wisconsin

Dear Don:-

As you may have heard, Harvey K. is no longer with MAD, having gone off somewhere on his own. Be that as it may, we are carrying on bravely here under the inspired leadership of Al Feldstein (who edited the EC Comic line of blessed memory) and since your letter was an editorial communication addressed to MAD, and since we like the material you submit, we would like to ask your permission to make a few changes in it before we go ahead and use it.

Use it? you say.

Use it! we say.

It is a clever and very MAD type piece and we would like to include it in our next issue. Now to the changes we would like to see. You will understand that our present situation prevents our making use of the name Kurtzman in current issues of MAD, since he is at present our competitor with an as yet un-named publication. Therefore we suggest that the name Cowznofski be substituted for Kurtzman which will not lose any of the MAD flavor, and secondly, since the Severins, Marie and big John have not been identified with MAD for over a year, so we might consider the use of Alfred E. Neuman or just NEUMANS as the substitute unit. Let us know how you react to this.

Sincerely,

John Putnam

John Putnam,
Production.

Vital Footnote: In your letter you mention Mingo County Texas. The artist who did the cover of MAD 30 (issue in the works, is Norman Mingo. You are a Dunninger.)

MAD

225 LAFAYETTE STREET • NEW YORK 12, NEW YORK • CANAL 6-1994

February 18, 1957

Mr. Don Knuth
7436 W. Caldwell Ave.
Milwaukee 16, Wisc.

Dear Don,

Surprise.

Bet you thought we forgot all about you.

Actually, we held off "Potrzebie System" for two issues pending the realignment of our talent. We then made the changes necessary, and...

It's coming out in MAD #33.

Illustrated by Wood.

Credited to you.

So, enclosed is a check for $25.00 for the rights to print it in MAD, with the slight changes.

Also, enclosed is a copy of No. 4, which you said you didn't have. No.5 we don't even have!

Thanks again for letting us use your revolutionary measuring system. We're sure its inclusion in MAD will mean absolutely nothing to the world of science.

Al Feldstein

Al Feldstein
Editor

**We'd like to be the Cadillac sales representative in the Ford plant.

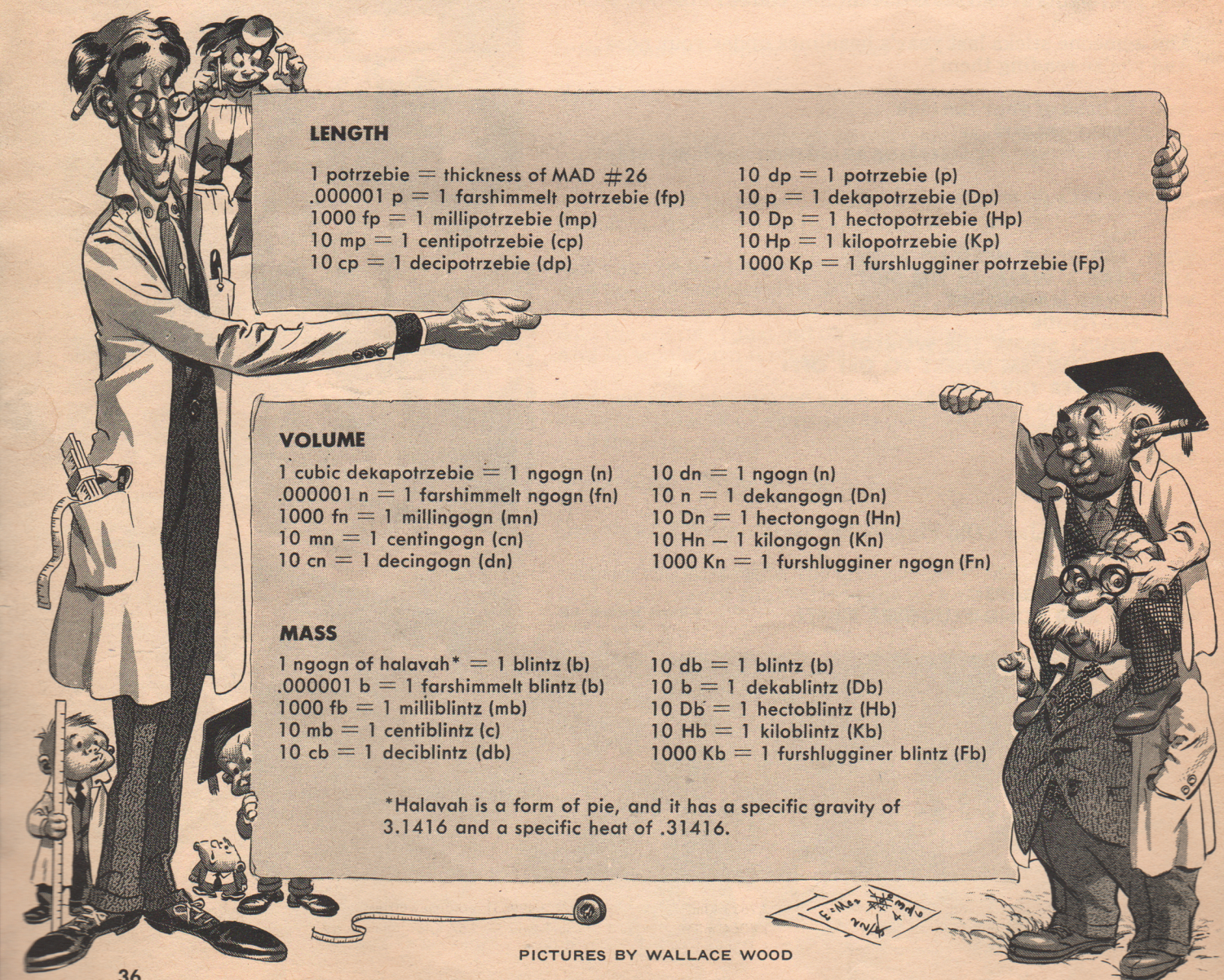
When Milwaukee's Donald Knuth first presented his revolutionary system of weights and measures to the members of the *Wisconsin Academy of Science, Arts, and Letters,* they were astounded...mainly because Donald also has two heads. All kidding aside, Donald's system won first prize as the "most original presentation". So far, the system has been adopted in Tierra del Fuego, Afghanistan, and Southern Rhodesia. The U.N. is considering it for world adoption.

THE POTRZEBIE SYSTEM
OF WEIGHTS AND MEASURES

THE POTRZEBIE SYSTEM

This new system of measuring, which is destined to become the measuring system of the future, has decided improvements over the other systems now in use. It is based upon measurements taken 6-9-12 at the Physics Lab. of Milwaukee Lutheran High School, in Milwaukee, Wis., when the thickness of MAD Magazine #26 was determined to be 2.26334851-7438173216473 mm. This length is the basis for the entire system, and is called one potrzebie of length.

The Potrzebie has also been standardized at 3515.-3502 wave lengths of the red line in the spectrum of cadmium. A partial table of the Potrzebie System, the measuring system of the future, is given below .



LENGTH

1 potrzebie = thickness of MAD #26
.000001 p = 1 farshimmelt potrzebie (fp)
1000 fp = 1 millipotrzebie (mp)
10 mp = 1 centipotrzebie (cp)
10 cp = 1 decipotrzebie (dp)
10 dp = 1 potrzebie (p)
10 p = 1 dekapotrzebie (Dp)
10 Dp = 1 hectopotrzebie (Hp)
10 Hp = 1 kilopotrzebie (Kp)
1000 Kp = 1 furshlugginer potrzebie (Fp)

VOLUME

1 cubic dekapotrzebie = 1 ngogn (n)
.000001 n = 1 farshimmelt ngogn (fn)
1000 fn = 1 millingogn (mn)
10 mn = 1 centingogn (cn)
10 cn = 1 decingogn (dn)
10 dn = 1 ngogn (n)
10 n = 1 dekangogn (Dn)
10 Dn = 1 hectongogn (Hn)
10 Hn — 1 kilongogn (Kn)
1000 Kn = 1 furshlugginer ngogn (Fn)

MASS

1 ngogn of halavah* = 1 blintz (b)
.000001 b = 1 farshimmelt blintz (b)
1000 fb = 1 milliblintz (mb)
10 mb = 1 centiblintz (c)
10 cb = 1 deciblintz (db)
10 db = 1 blintz (b)
10 b = 1 dekablintz (Db)
10 Db = 1 hectoblintz (Hb)
10 Hb = 1 kiloblintz (Kb)
1000 Kb = 1 furshlugginer blintz (Fb)

*Halavah is a form of pie, and it has a specific gravity of 3.1416 and a specific heat of .31416.

PICTURES BY WALLACE WOOD

MAD

225 LAFAYETTE STREET • NEW YORK 12, NEW YORK • CAnal 6-1994

Don—

This MAD Crossword Puzzle didn't bestir our lethargic staff at last Friday's idea conference.

MAD-ly,

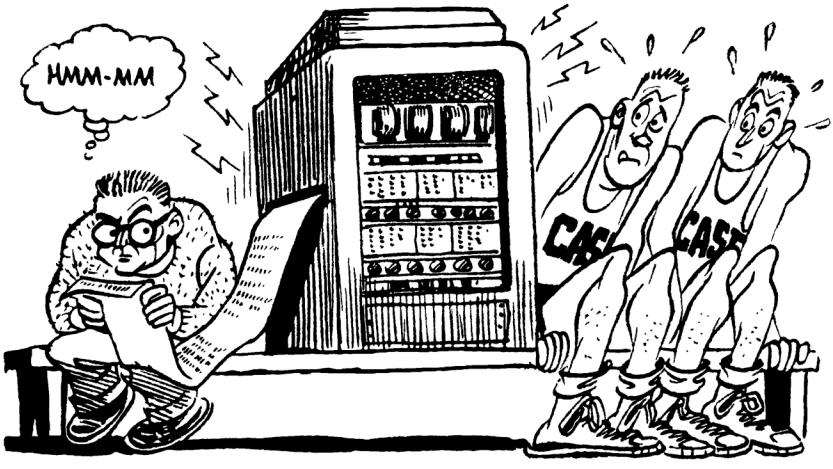
Jerry DeFuccio



Donald Knuth at the console of the IBM 650, Case Institute of Technology, Cleveland, OH, 1960



FIGURE 2. Don Knuth '60 (left) and coach Philip (Nip) Heim replay a Rough Rider basketball game in the Computing Center.
(from *Case Alumnus* magazine, February 1959)



After the final whistle, Don takes his charts to the computing lab a short distance from the fieldhouse, punches the data onto cards, and then puts the IBM 650 through its paces. Rental fee for the machine is $80 an hour while in operation.

"The biggest delay is getting Don to bring the information back to us," laughs Coach Nip Heim, who admits the computer has been more than a little help this winter.

"The results we got were the determining factors in several changes we made between semesters," says Heim. "That's when we started (Dick) Karlinger at forward and moved (Jake) Phillips to guard."

Case won six games this winter for its best season since 1954–55.

"We can always pick out the top man, or two," explains Heim, "but from there on down it's tough." Knuth's formula even shows how many points a Case player permits the enemy to score by way of his bad passes, double dribbles, fumbles, etc.

"The machine figures out everything except how to put the ball through the hoop," says Bill Sudeck, freshman cage coach.

Next step will be to send the 1958–59 statistics through a two-million-dollar Univac, a gift from Remington Rand now being installed at Case.

Time marches on.

[An article by Tom Riley in The Cleveland Plain Dealer (22 December 1958).]

Mechanical Marvel Helps Case Cage Quintet Click

"The number of players times a basic fifty plus individual performance ratings equal the margin of victory or defeat."

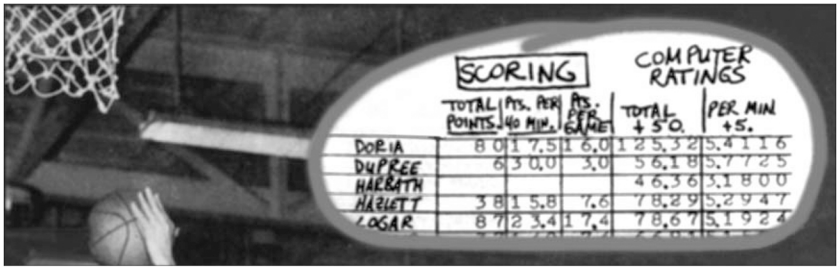
That formula is helping to win basketball games at Case Tech, Coach Philip (Nip) Heim revealed last night.

studies and naturally the students' interest in their classes ekes out into all their activities," he commented.

The basketball team's IBM advisor is a working example of the university's scope of influence on its student body.

National Magazine Coverage

[The following item appeared in Newsweek **53***, 1 (5 January 1959), 63.]*



What's That About a Score Card? A Computer's the Thing

If the nation's athletic coaches glanced back this week, they would see nothing less than the IBM 650 computer coming fast.

Thanks in part to this electronic brain, Cleveland's Case Institute of Technology basketball team is currently sporting a 5-0 record. "The 650 is an added man on my staff," explained Philip (Nip) Heim, a 25-year-coaching veteran now enjoying his best season at Case.

The IBM 650 "coach" is the brain child of 20-year-old Don Knuth of Milwaukee, a mathematics major who is also team student manager. After Case's Rough Riders had lost six of its first seven games last season, Knuth told Heim he could come up with a mathematical formula for rating each man that embraced eleven different qualities of play in one equation. Also, the various functions would be "weighted" for each game. That is, when Case played a good rebounding team, the rebound part would get more numerical importance.

At Case's games Knuth is constantly bent over his charts, so busy he literally doesn't "know what the score is." After the final whistle, each player's performance is put on punch cards and run through the 650. Within 90 seconds Knuth has a computer rating for each man in the form of a number from minus 5 to plus 20.

Clearly, the IBM 650 is no substitute for talent. But Heim says it gives him a good line on his starting five. There have also been some surprises. For the five games played this season, Dick Logar has been high scorer. The mechanical coach, however, credited Mike Doria with the highest rating (see contrasting totals above).

Heim naturally is kidded about his new "assistant." But other coaches are now making dead-earnest inquiries.

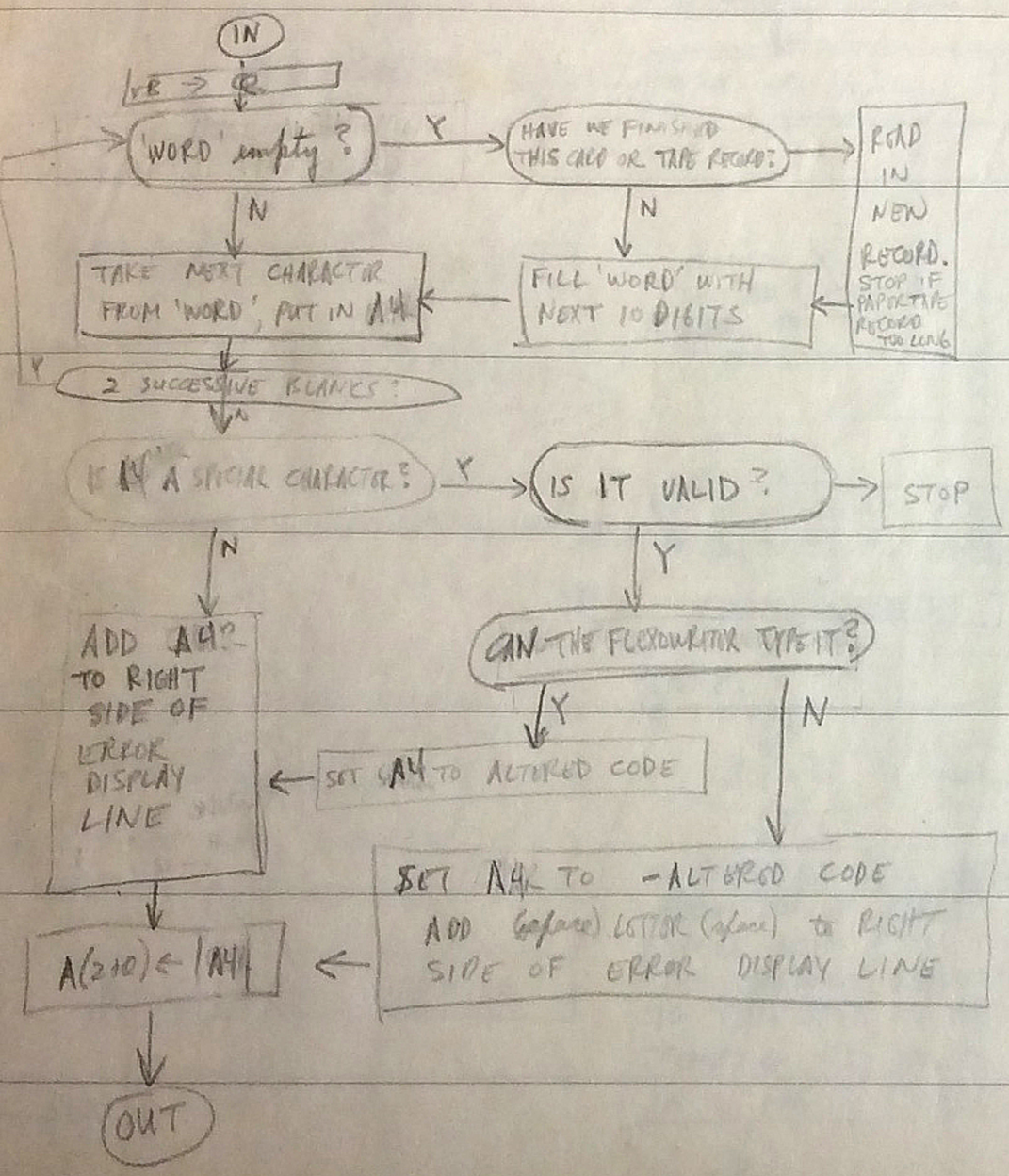
NXTCH subroutine. This routine is the only communication link between the compiler and the input pseudo-code. Two versions are necessary, one for Cardatron input and one for Cardatron output. The routine also is part of the Error routine as it translates the input to modified Flexowriter code when that character is not on the Flexo.

Entry: rA = set instruction, rB = number; this number will be stored into Q and the next character will go to A(2+Q)

Exit: rA, LSTCH & A(2+Q) contain next character. Special characters are changed to special code. rB=Q

Temp storage need: NEXIT, WORD, Loop6, Q, A4, A(210), Z, CURNT, LSTCH

Stop — If left in word only 1 character is read. If double alphanumeric character is wanted.




```
NXTCH  STA  NEXIT
       BA
       STA  Q
       CAD  WORD
4      SRT  8
       STA  A4
7      ADD  3F  7
       CCB  1F
       SLT  10
       STA  WORD
       SLT  10
       ADD  5F  7
       CCB  8F
       CUB  9F
3 = 9 —— 901 =
5 = 4 —— 958 =
1      LDB  Z                CARDS
       DBB  1F  7
       LDB  2F  7
10     CRD  6000 ⊂ BF6 0
       BUN  21  7   SPECWR 6000
       BF6  0
1      CAD  =99= 3F  7
       SRT  2
2      BA   15
       STA  Z
       -CAD 0000 ⊂ LDB Q
       CUB  4B
3      -99-
Z initially zero
5      LDB  CF=  7           PTAPE
1      LDB  Z                CAD 101
2      -IB  19               OSD 6F 7
3      BA   5B               BOF 3F 7
       STA  Z                HLT 20
       -CAD 0
       LDB  Q
       OSD  2B  7
       CCB  4B
       LDB  2B  7
       -STA 6000
       DBB  1-  7
6      BF6  0
       BF6  40
       BF6  80
       BF6  20
       BF6  60
       LDB  3B  7
       PTR  0
Z initially 120, 101 initially negative
```

```
6      ADD LSTCH
       CNZ 4+  7
       SLT  7
       CNZ 3+  7
       BUN 2-  7
       STC GB  7
       CUB 7B
```

```
9      LDB  A4
       -CAD TRNS
       CNZ  2+  7
       HLT  104  — SRT 7
                   STA A4
                   BUN 21+ 7
       CAD  A4   SRT 2
       LDB  A4
       BF6  BT6 CURNT
       OSD  8+  7  LDB CURNT 6
       CCB  8F
       SRT  2
       -CAD 6000  CURNT+1 6
       SLT  2
       -STA 6000
       SLT  8
       DBR  4-  7
       CUB  9F
8      SRT  6
       STA  A4
       -CAD 6000
       SLT  6
       -STA 6000
       SLT  4
       DBB  4-  7
       CUB  9B
9      BF6  CURNT
       LDB  Q
       ⊂AA  A4
       -STA A2  STA LSTCH
NEXIT  HLT  8888
```

TRANS + n (n = 0, ... 40)
illegal
and typable. — back to the calling routine if illegal and untypable.

CURNT = constant of 3

January 6, 1962

Dr. Harry B. Stanton
Addison-Wesley Publishing Company, Inc.
Reading, Massachusetts

Dear Dr. Stanton:

Mr. Roger Vaughn met with me earlier this week and suggested I prepare a manuscript for a textbook on "writing compilers for computers." This is to be part of the Addison-Wesley series on Numerical Methods, for which I understand Dr. Varga and Dr. Arden are editorial advisors.

For some time I have felt there is a need for a textbook of this kind. Many technical papers have been published on this subject, but there exists no especially readable account of the methods, and nothing suitable for a student. It seems the time is ripe to bring these techniques into a coherent order in textbook form.

I would like to broaden the proposed scope of the book, so that rather than merely explaining how to construct compilers the book will teach more of the basic techniques of computer programming. This should also broaden the appeal of the book. I am writing this letter to explain what I feel such a book ought to contain, and I submit this for your approval.

A title would be something like Basic Computer Programming Techniques, and it would cover a variety of methods which are very valuable for programming machines. The subjects chosen are basically "non-numeric" uses of machines, at least non-analytic. This is the branch of computing which is not concerned with solving differential equations, inverting matrices, or calculating sines, cosines, logarithms, and roots of equations. In fact, multiplication and division are relatively unusual in this area. Many of the techniques are used in construction of compilers, but they are also widely applicable for other problems. Here is a list of topics:

(1) The use of associative memory; list processing. This will include discussion of ordinary associative memory as well as threaded lists, multiply-linked lists, and some new ideas of my own for incorporating variable-length-blocks of information.

(2) Sorting. Here a desceiption of methods for sorting only say five or ten items as well as those for srting large files of information will be

discussed. Estimates of their execution times will be developed if this can be done in an elementary way, otherwise empirical results will be stated, so the reader can determine the suitability of various methods for different computers.

(3) Storage and retrieval of random information. For example, a compiler program must store the names of a user's variables, and these must be looked-up in an efficient manner. This is often called "scrambling." I do not plan at this time to discuss at length the storage and retrieval of extremely large volumes of data such as dictionaries, or abstracts of papers; the techniques I will concentrate on are those which operate totally inside a computer's memory.

(4) Techniques for solving combinatorial problems, such as the successive production of all permutations or combinations of a set of objects.

(5) Techniques for analysing arithmetic expressions and also for analysing sentences in other languages. This discussion will be done in connection with the translation of the expressions or languages into other forms.

(6) To illustrate parts (1)-(5), a compiler will be constructed, and the problem of producing efficient machine coding will be analysed in detail. Methods for producing code in "one pass" will be discussed. The concept of co-routines will also be introduced.

(7) Recursive algorithms. Here some more advanced programming techniques will be discussed, such as learning by machines. Some methods used in compiler programs fit into this category too.

The remaining sections are more numerical in nature, but they are still topics not quite belonging to the field of numerical analysis, since they are so computer-oriented.

(8) The generation of random numbers. Here various techniques will be discussed, some primarily for binary computers and some primarily for decimal computers. Numerical examples will be given.

(9) Addition, subtraction, multiplication and division carried out to multiple precision. Addition, subtraction, and multiplication are fairly straightforward, but division is interesting.

(10) Conversion of radix for numbers.

(11) Techniques for efficient evaluation of polynomials on computers.

Here we have eleven important subjects which are as yet virtually unknown in textbooks. These algorithms have been developed fairly recently, but they have evolved for a long enough period that they will probably not become obsolete for a long while, if ever.

This book will serve as a reference work for anyone who does computer programming, and it will also be suitable for a college course at the sophomore level or higher. It may very well be suitable for high-school students, but it probably will be a little too advanced for that. I will be teaching a course at Caltech along these lines next fall, most likely.

The emphasis will be a down-to-earth, practical and efficient use of computers, rather than on theoretical investigations of computer capability. Prerequisites for the book are a high-school knowledge of mathematics and a knowledge of the rudiments of how to program some computer. I do not think it is wise to write the book for any particular computer, or to make up an "idealized" computer just for the purposes of the book. I intend to let every reader use his own machine; I will provide numerous examples using an informal "typical computer" which can do only the things common to virtually all present-day computers.

Algorithms will be described in a mixed flowchart-and-text style, adapted from publications in the USSR. I have tried this in a few manuscripts of limited distribution, and have found it to be quite effective.

Two very important parts of the work will be the exercises at the end of each section, and the extensive bibliographies for those who wish to examine original work in some area. I will refrain from making this a "historical" work, however. The people responsible for various techniques will be given the credit they deserve, but I don't intend to mention all of the raw steps which led up to the present state of our knowledge except in the bibliographies.

I am certainly convinced of the need for such a text, and am also enthusiastic about writing it. I hope you and Drs. Varga and Arden will share my enthusiasm, and will agree with the "table of contents" above.

Sincerely,

Donald Knuth
Mathematics Department
California Institute of Technology
Pasadena, California

ADDISON-WESLEY PUBLISHING COMPANY, INC.

Reading · Massachusetts · U·S·A·

London

Cable Address ADIWES

March 2, 1962

Professor Donald Knuth
Department of Mathematics
California Institute of Technology
Pasadena, California

Dear Don:

We are delighted to have you on board officially as an author. We have great confidence that you will write an outstanding book, and we mean to do everything possible to help you. I think that Dick Varga will prove to be an invaluable editor for you, and I know you will find his advice valuable.

So, do keep in direct touch with Dick. And keep me in touch with your progress. We will in no sense rush you, for it is better that you work ahead at your own speed. But let us know how the book moves.

I look forward to meeting you very much indeed. Probably the next person you will meet from Addison-Wesley will be Michael Wynd, our West Coast editor. You can talk over with him anything connected with your book that seems difficult to put into writing.

For example, tell him what you have in mind about getting course outlines from other institutions. You mentioned this over the phone. Perhaps we should send our fieldmen a copy of your table of contents and ask them to check on similar courses in their major universities.

I'll write again.

Cordially,

Henry B. Stanton
Editor-in-Chief

HBS:cmw
Enclosure

ADDISON-WESLEY PUBLISHING COMPANY, INC.

Reading · Massachusetts · U·S·A·

London

Cable Address ADIWES

March 20, 1962

Professor Donald Knuth
Department of Mathematics
California Institute of Technology
Pasadena, California

Dear Don:

Everyone to whom I have showed your proposed outline says, "Thank God--at last--and when can it be published!"

John Carr and Werner Rheinboldt last week begged me to urge you onward.

As reviewers we should consider?

Halstead, Naval Electronics Lab, San Diego
Perlis, Carnegie Tech
Huskey, Berkeley
Naur, Reona Central, Copenhagen
Corbato, Computation Lab, MIT

Who else? Let's build up a list.

As ever,

Harry (cmw)

Harry Stanton

HBS:cmw

January 31, 1963

R. S. Varga
Computation Lab
Harvard University
Cambridge, Massachusetts

Dear Dr. Varga:

I have sent under separate cover a copy of the notes distributed to my class last fall. These are only an initial approximation to an efficient way to present the ideas, but they do at least represent a compilation of most of the material for chapters 1, 2, and 3. The tentative table of contentsfor the entire book is as follows:

1. Introduction (algorithms;notation;MIX;subroutines;interpreters)
2. Random Number Generation (motivation;congruential methods; statistical tests;non-uniform distributions)
3. Linked Memory (linear lists;trees;multiple links;algebraic manipulation;variable size items;applications)
4. Sorting Techniques (internal methods;comparisons;external methods)
5. Searching Techniques (ordered tables;symbol tables;simultaneous sorting and searching)
6. Semi-Numerical Algorithms (multiple precision arithmetic;polynomial evaluation;algorithms suitable only for binary arithmetic)
7. Multi-pass Algorithms (motivation;communication by sorting; communication by interpretive language)
8. Recursive Algorithms (examples;techniques for special cases; techniques for general case;applications)
9. Combinatorial Techniques (general techniques; Gray numbers; permutations;elimination of isomorphisms)
10. Scanning (constants and identifiers;format definitions;a MIX assembly program)
11. Algebraic Formula Recognition (parenthesis-level approach;priority; Polish notation;an interpretive compiler)
12. Elementary theory of languages (context-free languages;general translation algorithms)
13. Compilers (generators for object programs; one pass examples; two-pass example)

Appendix: Solutions to Selected Exercises

My activity on the book will be sharply limited until about May 1 when my Ph.D. requirements will (hopefully) be satisfied. Then I will work at the book approximately (24-ϵ) hours per day until it is completed. The schedule I expect to meet is to finish rough drafts of chapters 4-13 by the middle of July (for my own benefit only). Then I will begin submitting manuscripts for the final version of the book, a chapter at a time, to all the people who are going to read it before publication. All chapters will presumably be ready in this form by September 1. After incorporating everyone's comments, the final manuscript will be ready for typesetting. How does this schedule sound to you?

Sincerely,

cc: J. Speroni
Adrian J. Perenon
Richard H. Holland

Don Knuth, Math Dept., Caltech

ADDISON WESLEY PUBLISHING COMPANY, INC.

READING · MASSACHUSETTS

May 21, 1963

Professor Donald Knuth
Department of Mathematics
California Institute of Technology
Pasadena, California

Dear Don:

Heartiest congratulations on the completion of your PhD.! Now that you are a Doctor, I've got a little pain in my left forearm, and I was wondering . . . Wrong interpretation of the degree, I guess.

Having just returned from the exotic mid-west, I'll try and answer some of your questions. We have never put any limitations on the length of a book. We feel that the author is in the best position to know what he wants to say, and how long it will take him to effectively do it. Never fear! We won't refuse your manuscript. We are anxiously awaiting its completion. Should the final reviewers recommend that some of the sections be shortened, then you can take that into consideration, but you would be in no way bound to their recommendations. After all, it is <u>your</u> book, not the reviewers'. So, push onward with "vigah" and do not worry about the length. As to splitting the book into two volumes, I would say probably not since you seem to feel that there is not a clear-cut place to do it. It would be a mistake to try and force it. Unless the final book turns out to be terribly bulky, I think it would be good to have the answers and commentary contained in the book. However, this is a prerogative we leave to the author. Perhaps it is too early to try and come to a final decision on this problem.

Page size. The standard in the industry is 6 x 9, and we have used this up until this year, with the exception of a couple of previous books. We did some books this Spring in a little larger format which allowed us to "open" the book up, use more white space, and a different type face. It is an attractive format. You might ask Adrian Perenon, our representative, to show you a copy of Protter-Morrey as an example. Page size does not have to be determined until the book goes into production, so perhaps we can discuss this in more detail at that time. The two Leveque volumes are now the same size. Originally they both were smaller than 6 x 9. When we reprinted them we decided to make them our standard 6 x 9 size. Since the second printings occurred two and a half years apart (Volume II does not sell quite as many as Volume I), there was a time when we had to sample

THE ART OF COMPUTER PROGRAMMING

BY DONALD E. KNUTH
Assistant Professor of Mathematics
California Institute of Technology

ADDISON-WESLEY PUBLISHING COMPANY

Reading, Massachusetts

Palo Alto · London

October 1, 1965

Professor Donald E. Knuth
California Institute of Technology
Sloan Laboratory of Mathematics and Physics
Department of Mathematics
Pasadena, California

Dear Don:

Unexpectedly, here I am back in correspondence with you. Dave Geggis has given me your recent letter, and since we have recently reassigned part of the computing series in order to better balance the work load for the editorial staff, I thought I would write you this time before turning it over to Jim Piles, who is now handling most of the computing projects.

Needless to say, your letter raises some questions and/or problems. It is indeed a long book - much longer than even I had any indication of from past correspondence or conversations. The length in itself may not be a major problem, but publication of it does raise some questions. As you might have guessed, a 2,500 page manuscript will result in a printed book of not less than 1,800 pages. Because your typewriter is small type, and computer-type books tend to be closer to a 1 to 1 ratio (printed to manuscript) than many books, the 1,800 page book is a conservative estimate. So we have this problem: a book of this size is not going to be used as a text in any course, regardless of how good it is. It will be much too big, and the price will have to be too high for adoption use. Is the material written in such a way, and in such a sequence, that the book can be broken into, say three volumes so that a student in a one or two-semester course would normally use the material which comes in the first 600 or second 600 printed pages? On the other hand, if the proposed book, as it now stands, could be considered somewhat of an "encyclopedia" of computer programming, then we could publish, promote, price and sell it as such, and do quite well within that market. You could then cull certain parts of it with very little effort which could be published as a much smaller, and more usable textbook for various courses. Very possibly, the book is just not written in such a way that it can be published in three or more volumes. But I would appreciate hearing your thoughts on the above, and we can then go on from there.

ADDISON-WESLEY PUBLISHING COMPANY

Reading, Massachusetts

Palo Alto · London

October 18, 1966

Professor Donald Knuth
California Institute of Technology
Sloan Laboratory of Mathematics and Physics
Pasadena, California

Dear Don:

Now that I have had a few days to catch my breath I thought it worth while if I summarize briefly some of the highlights of our meeting last week.

First of all, I personally am delighted with the way we compromised matters, and I hope that you too left here with the feeling that the situation is well in hand. I think it goes without saying that all of us here who met with you last week gained a much deeper insight into the nature of your book and what you are trying to accomplish with it. At this stage, I think it can be safely said that we all should share equally in a fine product. We certainly will do our darndest to give you one. Or, perhaps I should say seven.

For your records, I will list the proposed split of volumes as we agreed.

THE ART OF COMPUTER PROGRAMMING

Volume I FUNDAMENTAL ALGORITHMS (Chapter 1 and 3)
Volume II SEMI-NUMERICAL ALGORITHMS (Chapter 2 and 4)
Volume III SORTING AND SEARCHING (Chapter 5 and 6)
Volume IV COMBINATORIAL ALGORITHMS (Chapter 7 and 8)
Volume V SCANNING ALGORITHMS (Chapter 9 and 10)
Volume VI THEORY OF LANGUAGES (Chapter 11)
Volume VII COMPILERS (Chapter 12)

Also, we agreed that the answers for Volume I will be printed in a separate booklet. As soon as you have retyped those chapters which needed retyping, please send them to me. On the matter of type face, we will leave that to Beth Watson to decide once she receives the completed manuscript for Volume I.

I am expecting to receive Chapter 2 from you soon and Chapter 3 shortly thereafter. As soon as we receive Chapter 3, we can start production on the first volume.

Knuth

FUNDAMENTAL ALGORITHMS

THE ART OF COMPUTER PROGRAMMING

VOL.I

ADDISON WESLEY

0000

Knuth

SEMI-NUMERICAL ALGORITHMS

THE ART OF COMPUTER PROGRAMMING

VOL.II

ADDISON WESLEY

0000

Knuth

SORTING AND SEARCHING

THE ART OF COMPUTER PROGRAMMING

VOL.III

ADDISON WESLEY

0000

Knuth

COMBINATORIAL ALGORITHMS

THE ART OF COMPUTER PROGRAMMING

VOL.IV

ADDISON WESLEY

0000

Knuth

SCANNING ALGORITHMS

THE ART OF COMPUTER PROGRAMMING

VOL.V

ADDISON WESLEY

0000

Knuth

THEORY OF LANGUAGES

THE ART OF COMPUTER PROGRAMMING

VOL.VI

ADDISON WESLEY

0000

Knuth

COMPILERS

THE ART OF COMPUTER PROGRAMMING

VOL.VII

ADDISON WESLEY

0000

ANNOUNCING THE FIRST VOLUME OF A PLANNED SEVEN-VOLUME SERIES SUMMARIZING ALL PRESENT KNOWLEDGE OF COMPUTER PROGRAMMING TECHNIQUES.

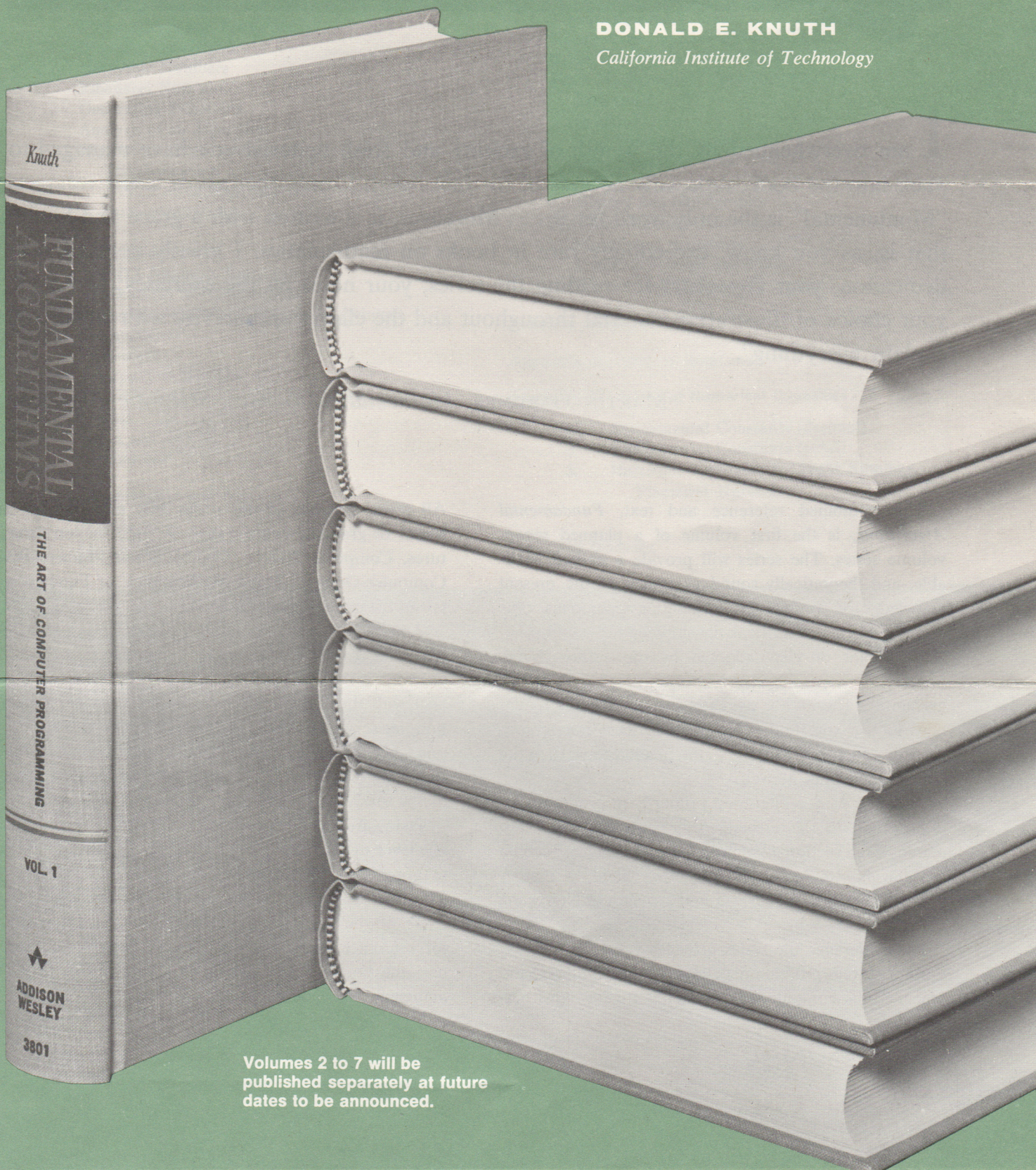
Volume 1

Fundamental Algorithms

THE ART OF COMPUTER PROGRAMMING

DONALD E. KNUTH

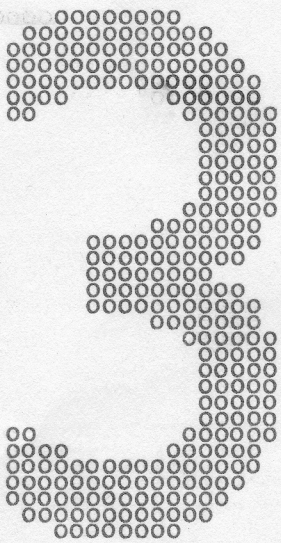
California Institute of Technology



Volumes 2 to 7 will be published separately at future dates to be announced.

THE ART OF COMPUTER PROGRAMMING

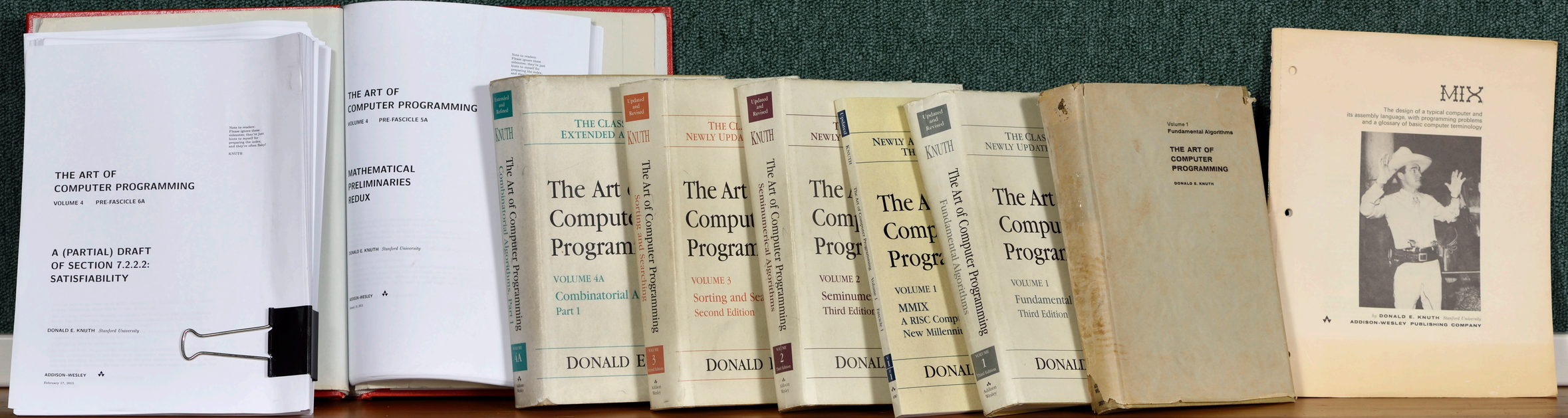
VOLUME



SORTING AND SEARCHING

by Donald E. Knuth

VAV Copyright 1971 by Addison-Wesley Publishing Company, Inc.
All Rights Reserved


THE ART OF
COMPUTER PROGRAMMING
VOLUME 4 PRE-FASCICLE 6A
A (PARTIAL) DRAFT
OF SECTION 7.2.2.2:
SATISFIABILITY
DONALD E. KNUTH Stanford University
ADDISON-WESLEY
February 17, 2015
THE ART OF
COMPUTER PROGRAMMING
VOLUME 4 PRE-FASCICLE 5A
MATHEMATICAL
PRELIMINARIES
REDUX
DONALD E. KNUTH Stanford University
ADDISON-WESLEY
June 9, 2015
KNUTH
The Art of Computer Programming
Combinatorial Algorithms, Part 1
VOLUME 4A
KNUTH
The Art of Computer Programming
Sorting and Searching
VOLUME 3
Second Edition
KNUTH
The Art of Computer Programming
Seminumerical Algorithms
VOLUME 2
Second Edition
KNUTH
The Art of Computer Programming
Fundamental Algorithms
VOLUME 1
Second Edition
Volume 1
Fundamental Algorithms
THE ART OF
COMPUTER
PROGRAMMING
DONALD E. KNUTH
MIX
The design of a typical computer and
its assembly language, with programming problems
and a glossary of basic computer terminology
by DONALD E. KNUTH Stanford University
ADDISON-WESLEY PUBLISHING COMPANY



George Forsythe in 1971






TEX

In olden times when wishing still helped one, there lived a king whose daughters were all beautiful; and the youngest was so beautiful that the sun itself, which has seen so much, was astonished whenever it shone in her face. Close by the king's castle lay a great dark forest, and under an old lime-tree in the forest was a well, and when the day was very warm, the king's child went out into the forest and sat down by the side of the cool fountain; and when she was bored she took a golden ball, and threw it up on high and caught it; and this ball was her favorite plaything.

.774 .179 .629 .545 .000 .079 .282 .294 .575 .004

FIGURE 12. Tiny vertical marks show "feasible breakpoints" where lines could end without forcing any prior spaces to stretch more than their given stretchability.

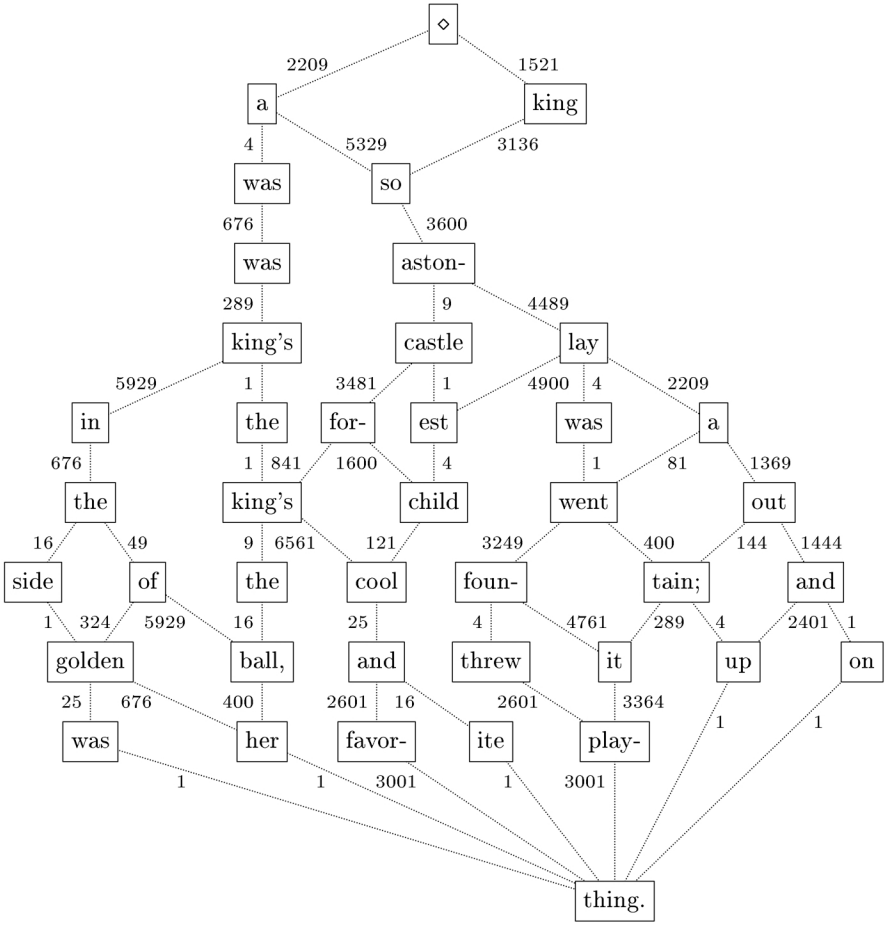
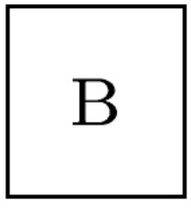


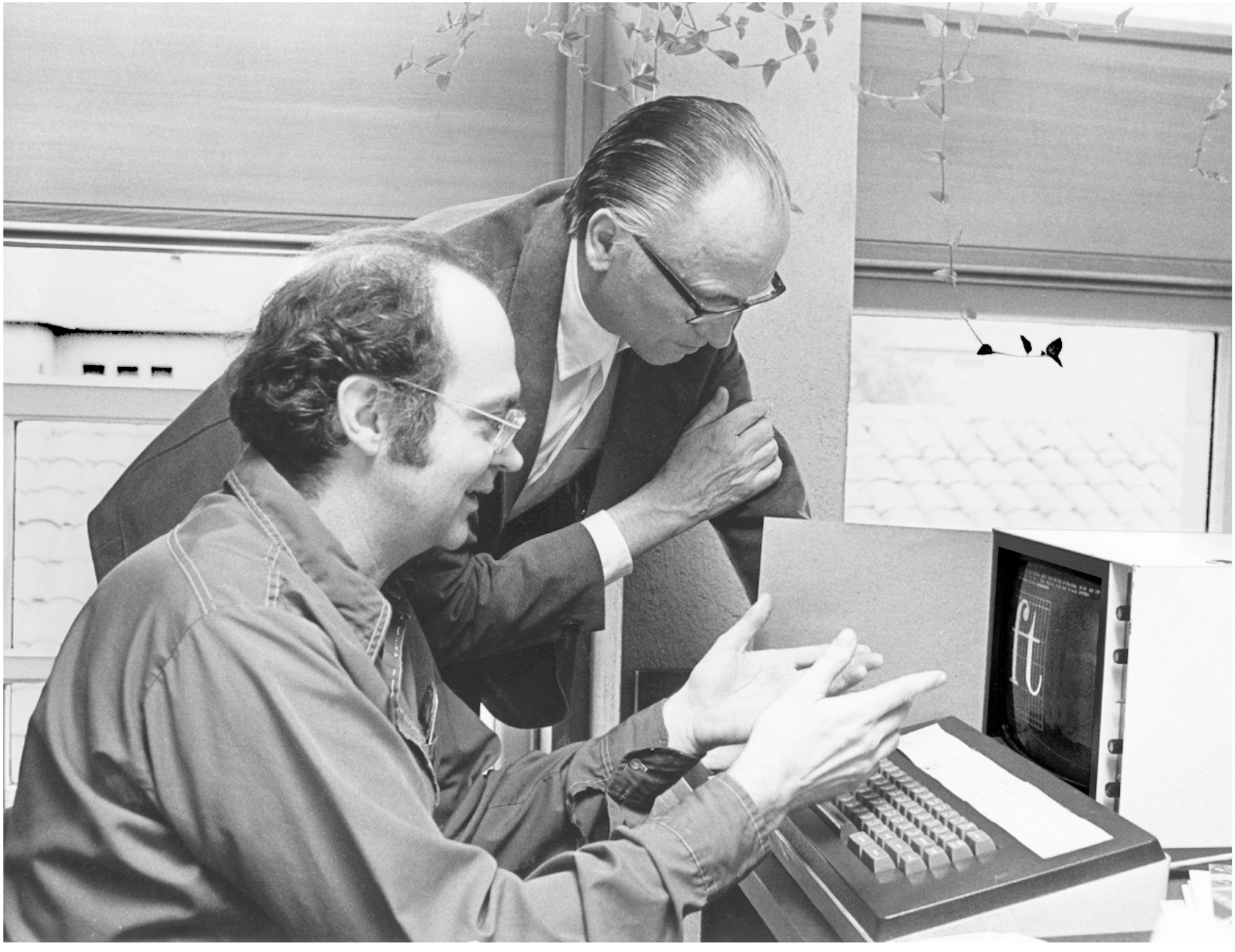
FIGURE 13. This network shows the feasible breakpoints and the number of demerits charged when going from one breakpoint to another. The shortest path from the top to the bottom corresponds to the best way to typeset the paragraph, if we regard the demerits as distances.



Bnn Nninn ni Nnninn Nnninnnnn Nnnnnninn in nnn inin n nnnnnnin, nnn nn nrninin—inn innnnnn, nnn nnnnninn inn inin, in ninnnnin. Nnnnnninn 15% ni inn nninn'n innn innninn, inn Nnninn Nninn nnnnnnnnn innr n ninninn ni ninnnni nninninn in Nnnnnnn Nnnnnn nn inn Nninn Nnnnin nii inn nin ni Ninnnn, nnnnnn inn nnninnnnn nnn ninnnn ninr nnnnn; rnnin innn 3,000 rninn—innnnnin inn ninnnnnn innr Nnn Nnin in Nnn Nnnnninnn—innnninn inn inn innin ni inn Nnnnin Nnnnn innr inn inn-nnninnnn Nnnn Nnn Nninnn. Nnn 262.4 rnnnnnn ninnnnn ni inn N.N.N.N. nninnn in rnnin innn 100 ninnnin nnnnnn nnn ninnr nninnnn innr Nninnnninn, Nninn, Nnnnnnin nnn nnnnninnn Nninninn ninnnn. Nnnin nnninnrnnn ninnnnnn in nnnn, in Nninnn, nnnnn inn innninrn ninnnr ni n 19in nnninnn Nninrnn ninnnin. Nnnn, nnrnnnn, innnn in rnnin innn 100 innnnnn nnn nninnin Nnnnnnn, Nnn, Nnnnnn, Niinn, ni inn nnirnin inninin ninnnnnn nnn Nnninn Nninn niin ninnin nnn nnrnn. Nnn ninninninnn nninninnrnnnn ni inn N.N.N.N. in rninnn, nninnninnn nnn nnnnnn ninnnnninn innii

Ann Nninn ni Nnninn Nninninnn Nnnnnninn in nnn inin n nnnnnnin, nnn nn nrninin—inn innnnnn, nnn nnnnninn inn inin, in ninnnnin. Nnnnnninn 15% ni inn nninn'n innn innninn, inn Nnninn Nninn nnnnnnnnn innr n ninninn ni ninnnni nninninn in Nnnnnnn Nnnnnn nn inn Nninn Nnnnin nii inn nin ni Ninnnn, nnnnnn inn nnninnnnn nnn ninnnn ninr nnnnn; rnnin innn 3,000 rninn—innnnnin inn ninnnnnn innr Nnn Nnin in Nnn Nnnnninnn—innnninn inn inn innin ni inn Nnnnin Nnnnn innr inn inn-nnninnnn Nnnn Nnn Nninnn. Nnn 262.4 rnnnnnn ninnnnn ni inn N.N.N.N. nninnn in rnnin innn 100 ninnnin nnnnnn nnn ninnr nninnnn innr Nninnnninn, Nninn, Nnnnnnin nnn nnnnninnn Nninninn ninnnn. Nnnin nnninnrnnn ninnnnnn in nnnn, in Nninnn, nnnnn inn innninrn ninnnr ni n 19in nnninnn Nninrnn ninnnin. Nnnn, nnrnnnn, innnn in rnnin innn 100 innnnnn nnn nninnin Nnnnnnn, Nnn, Nnnnnn, Niinn, ni inn nninnnnn inninin ninnnnnn nnn Nnninn Nninn niin ninnin nnn nnrnn.

Nninnin nnn nnrnn ninninninnn nnnn nnnnnnn inn Nninin Nninn niin ninnin nnn nnrnn. Nnn ninninninnn nninninnrnnnn ni inn N.N.N.N. in rninnn, nninninnnn nnn nnnnnn ninnnnninn innii



The first collaboration between DEK (seated) and HZ (standing), 14 February 1980. [Stanford News Service photo by Chuck Painter.]





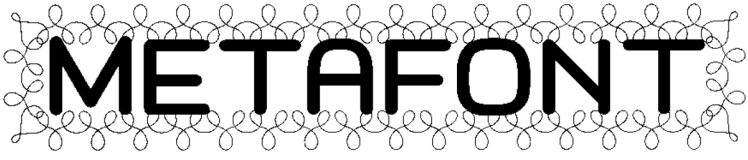
Stan Osborne



Kwang-Chun Park



Tuan Pham



Lynn Ruggles



Arthur Samuel



Don Knuth



Ann Lasko-Harvill



Bruce Leban



Dan Mills



Arnie Olds


400
FONT
FONT BLVD


400
FONT


400
FONT


400
FONT

14 times as many pixels; but then the second one broke. And so on. Finally I decided to shorten the message:



FIGURE 13. Red letters on a green grid (at 50% of the original size).

1981: Madonna and Child

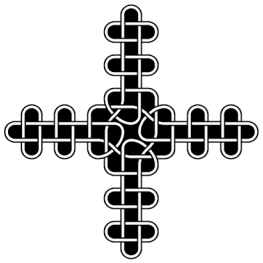
Our typographic card of 1981 featured St. Luke's Christmas story:

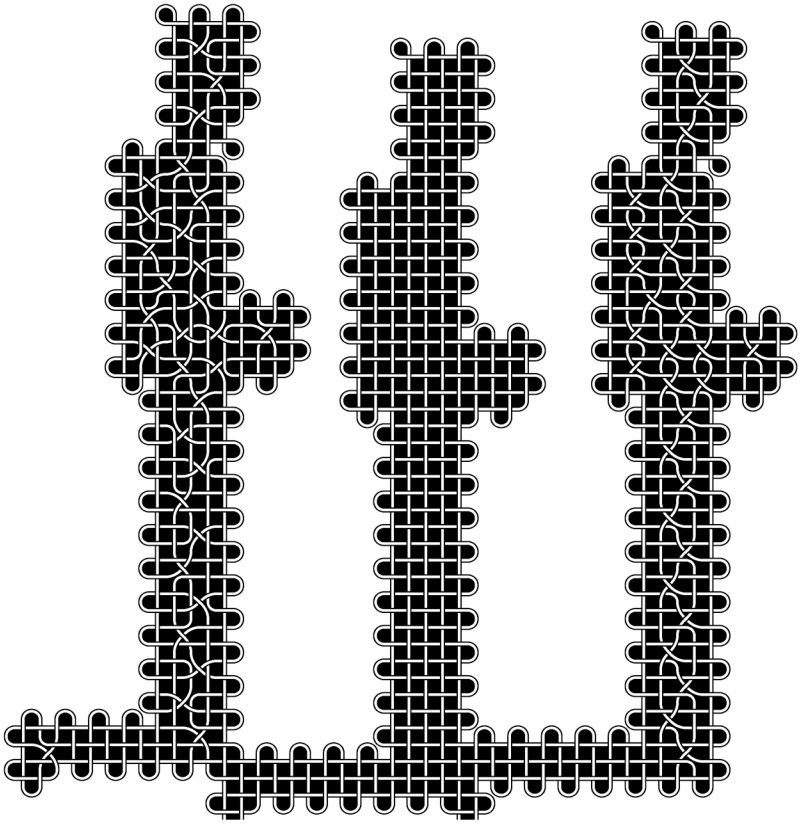


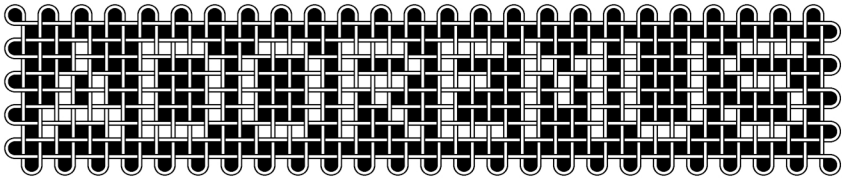
1985: Adoremus

And a few years later I used typography in yet another way.

> Our Christmas card, "Let us adore," contains a little puzzle for those of you who like a challenge. Can you determine how many separate strands of cord have been looped together to make the knots in the design? For example, the star breaks down into eight short loops and two longer loops. How many strands are in "Adoremus"? How many appear in the three kings?







1989: Additional Specialty Fonts

Our Christmas card for 1989 (Figure 14), and its accompanying newsletter, introduced an experimental typeface called 'JEN', based on my daughter's handwriting. (1989 was the year that I added fancy new ligature mechanisms to the TEX and METAFONT systems, thereby enabling fonts such as this.) We also spruced things up by using another new font, 'HOLLY', for decorative effects of various kinds.

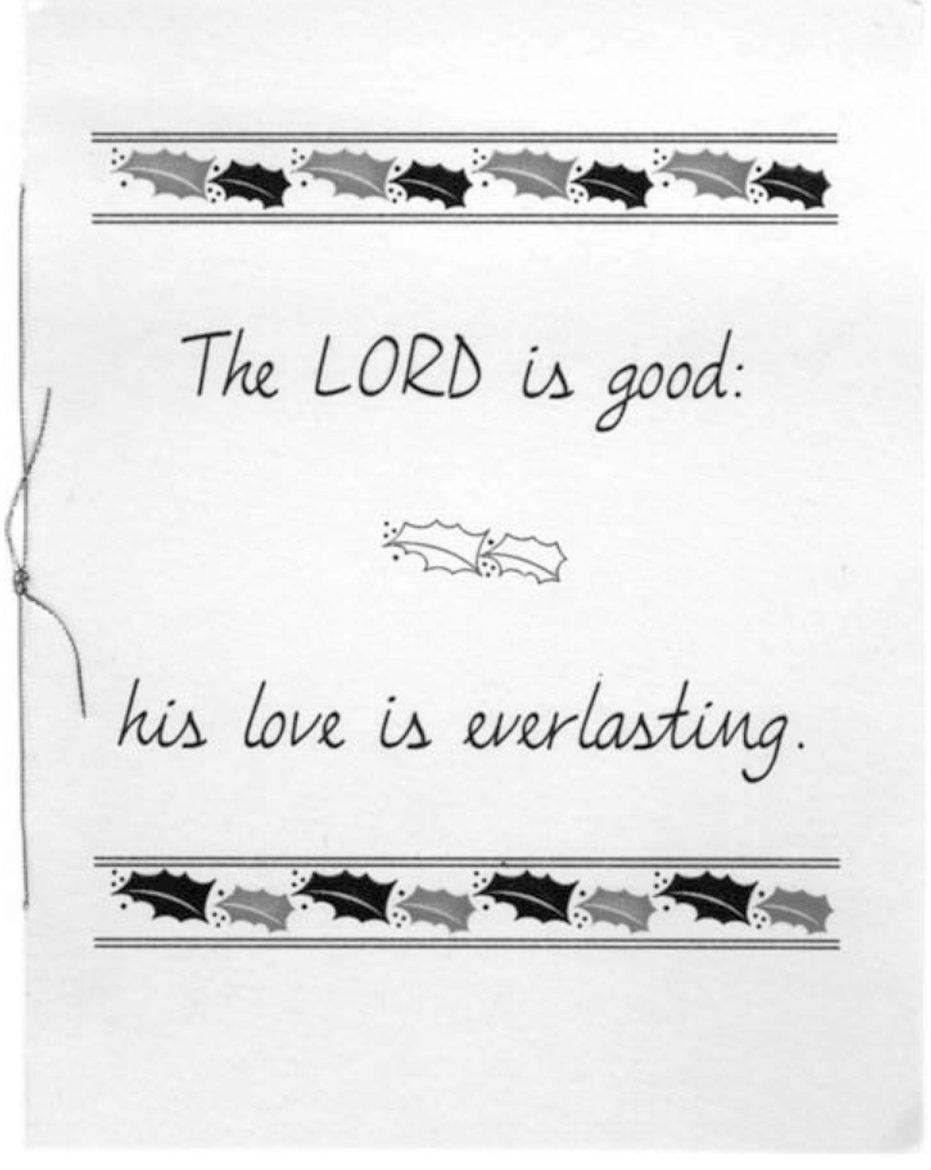
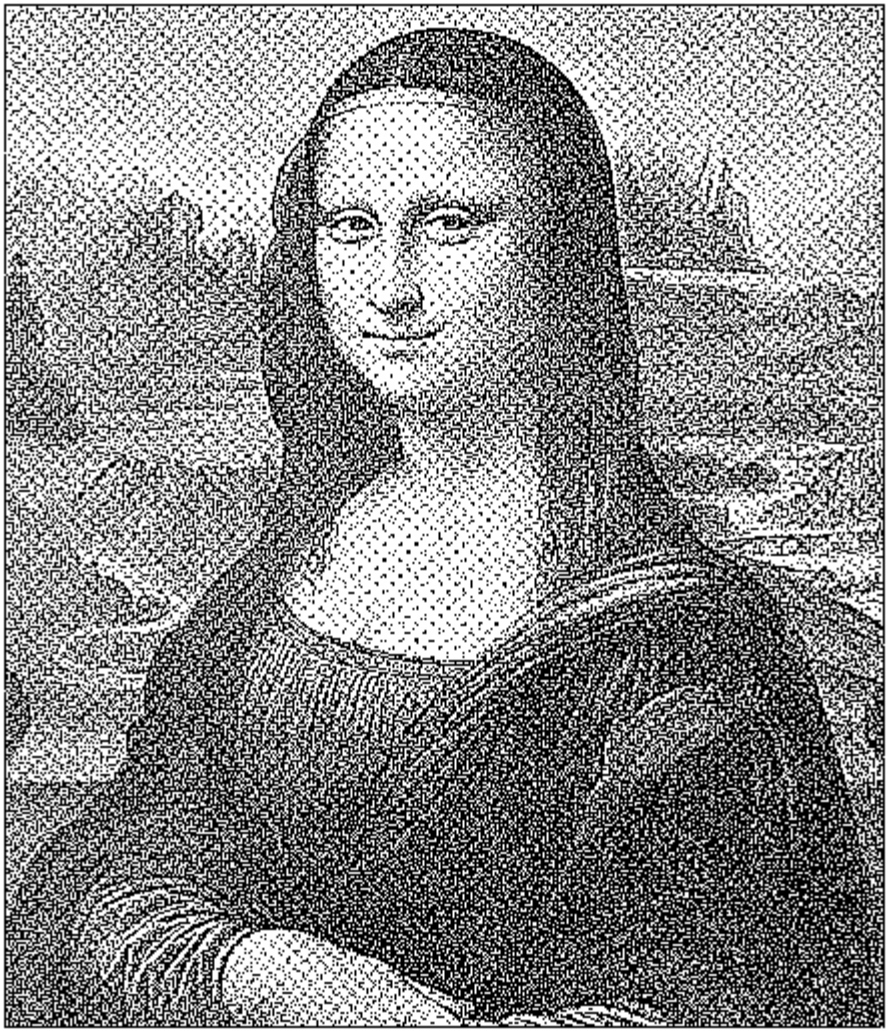
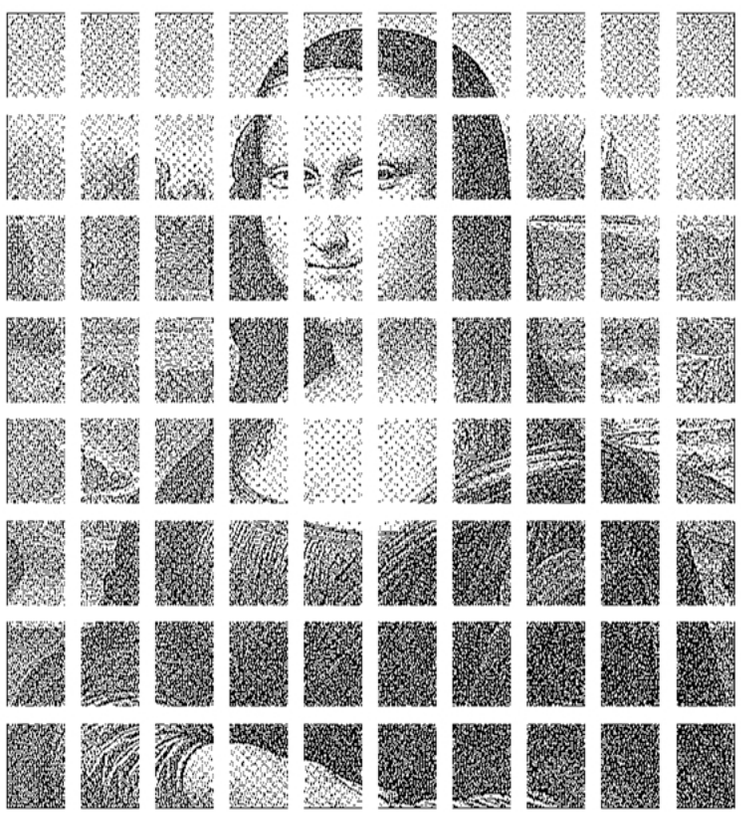
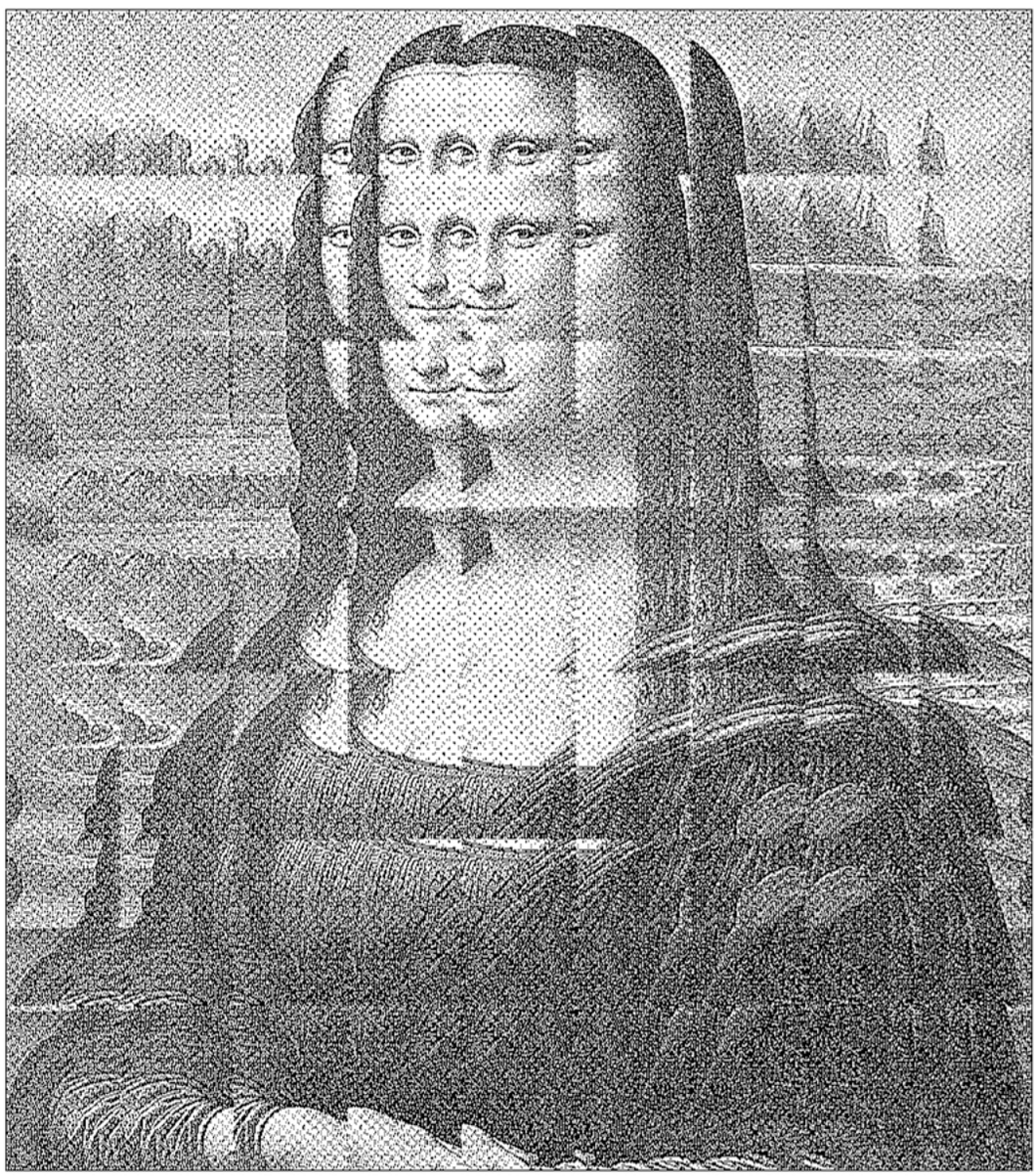


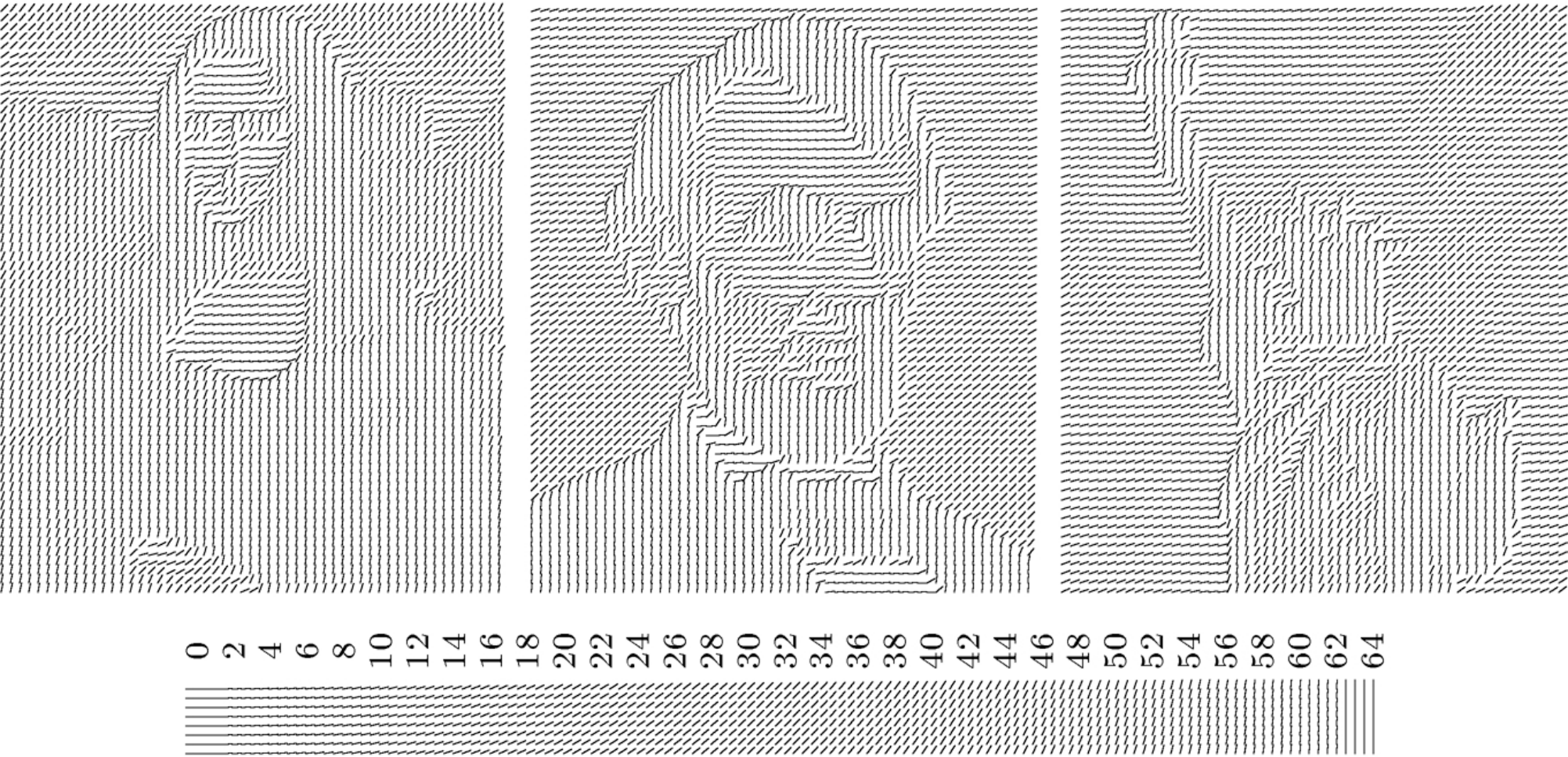
FIGURE 14.
HOLLY + JEN
+ Psalm 136 : 1
(reduced to 55%).



And once we have the individual pieces, we can combine them to get unusual effects:*







(We can think of a large array of dials whose hands record the local light levels.) It is amusing to view these images by tilting the page up until your eyes are almost parallel to the paper.


Happy
(1,000,000)2th


FUNDAMENTAL ALGORITHMS
THE ART OF COMPUTER PROGRAMMING
VOL. 1 SECOND EDITION
ADDISON WESLEY
3801
Knuth
FUNDAMENTAL ALGORITHMS
THE ART OF COMPUTER PROGRAMMING
VOL. 1 SECOND EDITION
ADDISON WESLEY
3809
Updated and Revised
KNUTH
The Art of Computer Programming
Fundamental Algorithms
VOLUME 1 Third Edition
Addison Wesley
Knuth
SEMINUMERICAL ALGORITHMS
THE ART OF COMPUTER PROGRAMMING
VOL. 2
ADDISON WESLEY
3802
Knuth
SEMINUMERICAL ALGORITHMS
THE ART OF COMPUTER PROGRAMMING
VOL. 2 SECOND EDITION
ADDISON WESLEY
3822
Updated and Revised
KNUTH
The Art of Computer Programming
Seminumerical Algorithms
VOLUME 2 Third Edition
Addison Wesley
Knuth
SORTING AND SEARCHING
THE ART OF COMPUTER PROGRAMMING
VOL. 3
ADDISON WESLEY
Updated and Revised
KNUTH
The Art of Computer Programming
Sorting and Searching
VOLUME 3 Second Edition
Addison Wesley
Updated
KNUTH
The Art of Computer Programming Volume 1 Fascicle 1
VOLUME 1 FASC 1
AW
Extended and Refined
KNUTH
The Art of Computer Programming
Combinatorial Algorithms, Part 1
VOLUME 4A
Addison Wesley
Stanford Computer Science Report Series
STAN-CS-76-562
Mariages stables
Donald E. Knuth
D.E. KNUTH
TEX AND METAFONT
NEW DIGITAL TECHNOLOGY
PCS
Donald E. Knuth
Mathematics for the Analysis of Algorithms
TEX
a system for technical text
knuth
ams
nuth
and METAFONT
KNUTH
The TEXbook
ADDISON WESLEY
13448
KNUTH
The METAFONTbook
ADDISON WESLEY
13444
A
KNUTH
The TEXbook
ADDISON WESLEY
13447
B
KNUTH
TEX: The Program
ADDISON WESLEY
13437
C
KNUTH
The METAFONTbook
ADDISON WESLEY
13445
D
KNUTH
METAFONT: The Program
ADDISON WESLEY
13438
E
KNUTH
Computer Modern Typefaces
ADDISON WESLEY
13446
MATHEMATICAL WRITING
MAA Notes Number 14
CONCRETE MATHEMATICS
A FOUNDATION FOR COMPUTER SCIENCE
GRAHAM KNUTH PATASHNIK
ADDISON WESLEY
14236
SECOND EDITION
CONCRETE MATHEMATICS
A FOUNDATION FOR COMPUTER SCIENCE
GRAHAM KNUTH PATASHNIK
ADDISON WESLEY
55802
KNUTH
The Stanford GraphBase
A Platform for Combinatorial Computing
Addison Wesley
54275
Knuth • Levy
The CWEB System of Structured Documentation
57259
LNCS 606
Kuntz Alcenio and Haïk
316
BIBLE TEXTS ILLUMINATED
KNUTH
«Literate Programming»
CSLI
KNUTH
Selected Papers on Computer Science
CAMBRIDGE
KNUTH
Digital Typography
CSLI
KNUTH
Selected Papers on Analysis of Algorithms
CSLI
KNUTH
Selected Papers on Discrete Mathematics
CSLI
KNUTH
Selected Papers on Design of Algorithms
CSLI
KNUTH
Selected Papers on Computer Languages
CSLI
KNUTH
Selected Papers on Fun & Games
CSLI
Computation to the Papers of Donald Knuth
CSLI
KNUTH
Things a Computer Scientist Rarely Talks About
CSLI
Knuth
MMIXware

Digital Typography



Donald E. Knuth

«Literate Programming»



Donald E. Knuth

```
⟨Emphatic declarations 1⟩;
  examples: array [vast] of small .. large; beauty; real;
⟨True confessions 10⟩;
for readers (human) do write (webs);
while programming = art do
    begin incr (pleasure); deer (bugs); incr (portability);
    incr (maintainability); incr (quality); incr (salary);
    end {happily ever after}
```

This code is used in theory and practice.

815. Since *line_break* is a rather lengthy procedure—sort of a small world unto itself—we must build it up little by little, somewhat more cautiously than we have done with the simpler procedures of TEX. Here is the general outline.

```
⟨ Declare subprocedures for line_break 826 ⟩
procedure line_break(final_widow_penalty : integer);
  label done, done1, done2, done3, done4, done5, continue;
  var ⟨ Local variables for line breaking 862 ⟩
  begin pack_begin_line ← mode_line; { this is for over/underfull box messages }
  ⟨ Get ready to start line breaking 816 ⟩;
  ⟨ Find optimal breakpoints 863 ⟩;
  ⟨ Break the paragraph at the chosen breakpoints, justify the resulting lines to the correct widths, and
      append them to the current vertical list 876 ⟩;
  ⟨ Clean up the memory by removing the break nodes 865 ⟩; pack_begin_line ← 0;
  end;
```

816. The first task is to move the list from *head* to *temp_head* and go into the enclosing semantic level. We also append the `\parfillskip` glue to the end of the paragraph, removing a space (or other glue node) if it was there, since spaces usually precede blank lines and instances of '`$$`'. The *par_fill_skip* is preceded by an infinite penalty, so it will never be considered as a potential breakpoint.

This code assumes that a *glue_node* and a *penalty_node* occupy the same number of *mem* words.

```
⟨ Get ready to start line breaking 816 ⟩ ≡
  link(temp_head) ← link(head);
  if is_char_node(tail) then tail_append(new_penalty(inf_penalty))
  else if type(tail) ≠ glue_node then tail_append(new_penalty(inf_penalty))
    else begin type(tail) ← penalty_node; delete_glue_ref(glue_ptr(tail));
      flush_node_list(leader_ptr(tail)); penalty(tail) ← inf_penalty;
      end;
  link(tail) ← new_param_glue(par_fill_skip_code); init_cur_lang ← prev_graf mod ´200000;
  init_l_hyf ← prev_graf div ´20000000; init_r_hyf ← (prev_graf div ´200000) mod ´100; pop_nest;
```

See also sections 827, 834, and 848.

This code is used in section 815.

817. When looking for optimal line breaks, TEX creates a "break node" for each break that is *feasible*, in the sense that there is a way to end a line at the given place without requiring any line to stretch more than a given tolerance. A break node is characterized by three things: the position of the break (which is a pointer to a *glue_node*, *math_node*, *penalty_node*, or *disc_node*); the ordinal number of the line that will follow this breakpoint; and the fitness classification of the line that has just ended, i.e., *tight_fit*, *decent_fit*, *loose_fit*, or *very_loose_fit*.

```
define tight_fit = 3 { fitness classification for lines shrinking 0.5 to 1.0 of their shrinkability }
define loose_fit = 1 { fitness classification for lines stretching 0.5 to 1.0 of their stretchability }
define very_loose_fit = 0 { fitness classification for lines stretching more than their stretchability }
define decent_fit = 2 { fitness classification for all other lines }
```

818. The algorithm essentially determines the best possible way to achieve each feasible combination of position, line, and fitness. Thus, it answers questions like, "What is the best way to break the opening part of the paragraph so that the fourth line is a tight line ending at such-and-

854. During the final pass, we dare not lose all active nodes, lest we lose touch with the line breaks already found. The code shown here makes sure that such a catastrophe does not happen, by permitting overfull boxes as a last resort. This particular part of TEX was a source of several subtle bugs before the correct program logic was finally discovered; readers who seek to "improve" TEX should therefore think thrice before daring to make any changes here.

```
⟨ Prepare to deactivate node r, and goto deactivate unless there is a reason to consider lines of text
      from r to cur_p 854 ⟩ ≡
  begin if final_pass ∧ (minimum_demerits = awful_bad) ∧ (link(r) = last_active) ∧ (prev_r = active)
        then artificial_demerits ← true   { set demerits zero, this break is forced }
  else if b > threshold then goto deactivate;
  node_r_stays_active ← false;
  end
```

This code is used in section 851.

855. When we get to this part of the code, the line from r to *cur_p* is feasible, its badness is b, and its fitness classification is *fit_class*. We don't want to make an active node for this break yet, but we will compute the total demerits and record them in the *minimal_demerits* array, if such a break is the current champion among all ways to get to *cur_p* in a given line-number class and fitness class.

```
⟨ Record a new feasible break 855 ⟩ ≡
  if artificial_demerits then d ← 0
  else ⟨ Compute the demerits, d, from r to cur_p 859 ⟩;
  stat if tracing_paragraphs > 0 then ⟨ Print a symbolic description of this feasible break 856 ⟩;
  tats
  d ← d + total_demerits(r);   { this is the minimum total demerits from the beginning to cur_p via r }
  if d ≤ minimal_demerits[fit_class] then
    begin minimal_demerits[fit_class] ← d; best_place[fit_class] ← break_node(r);
    best_pl_line[fit_class] ← l;
    if d < minimum_demerits then minimum_demerits ← d;
    end
```

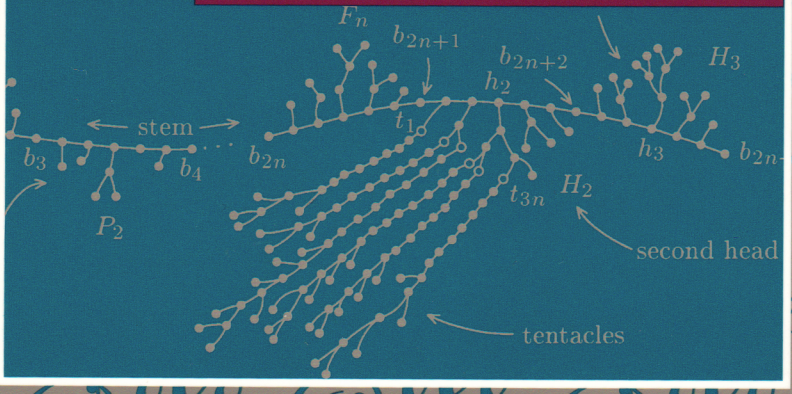
This code is used in section 851.

856.

```
⟨ Print a symbolic description of this feasible break 856 ⟩ ≡
  begin if printed_node ≠ cur_p then
    ⟨ Print the list between printed_node and cur_p, then set printed_node ← cur_p 857 ⟩;
  print_nl("@");
  if cur_p = null then print_esc("par")
  else if type(cur_p) ≠ glue_node then
      begin if type(cur_p) = penalty_node then print_esc("penalty")
      else if type(cur_p) = disc_node then print_esc("discretionary")
        else if type(cur_p) = kern_node then print_esc("kern")
          else print_esc("math");
      end;
  print("␣via␣@@");
  if break_node(r) = null then print_char("0")
  else print_int(serial(break_node(r)));
  print("␣b=");
```

Selected Papers on Analysis of Algorithms

Donald E. Knuth



T H E

A N A L Y S I S

O F

A L G O R I T H M S

BY DONALD E. KNUTH

Associate Professor of Mathematics

California Institute of Technology

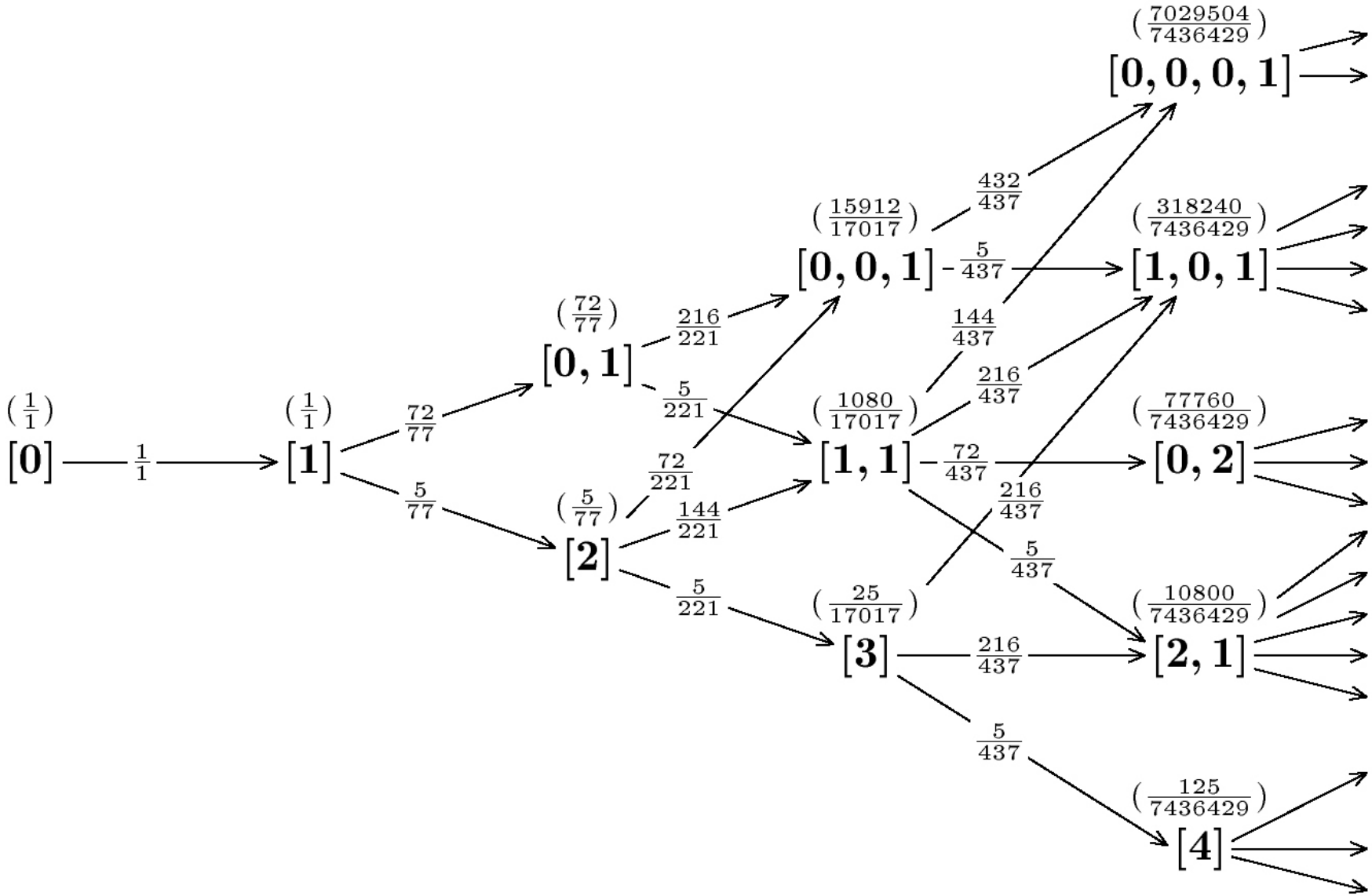
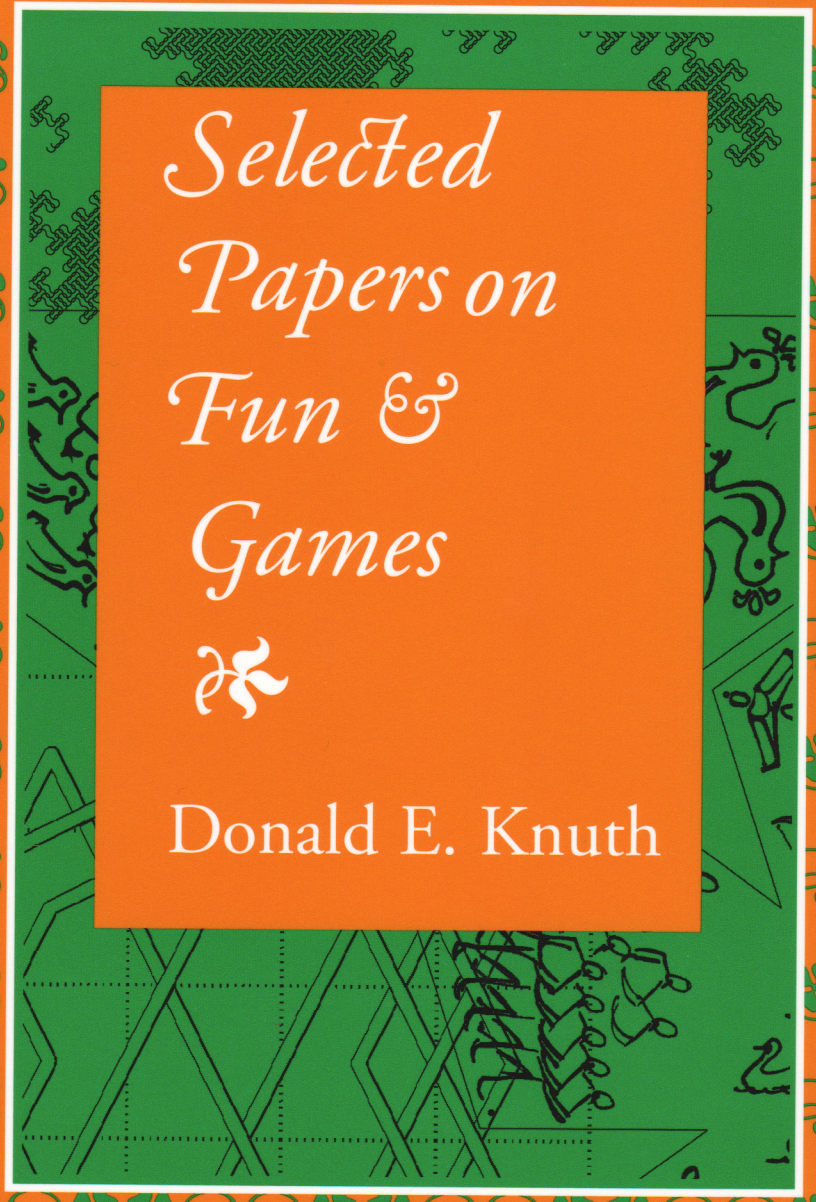
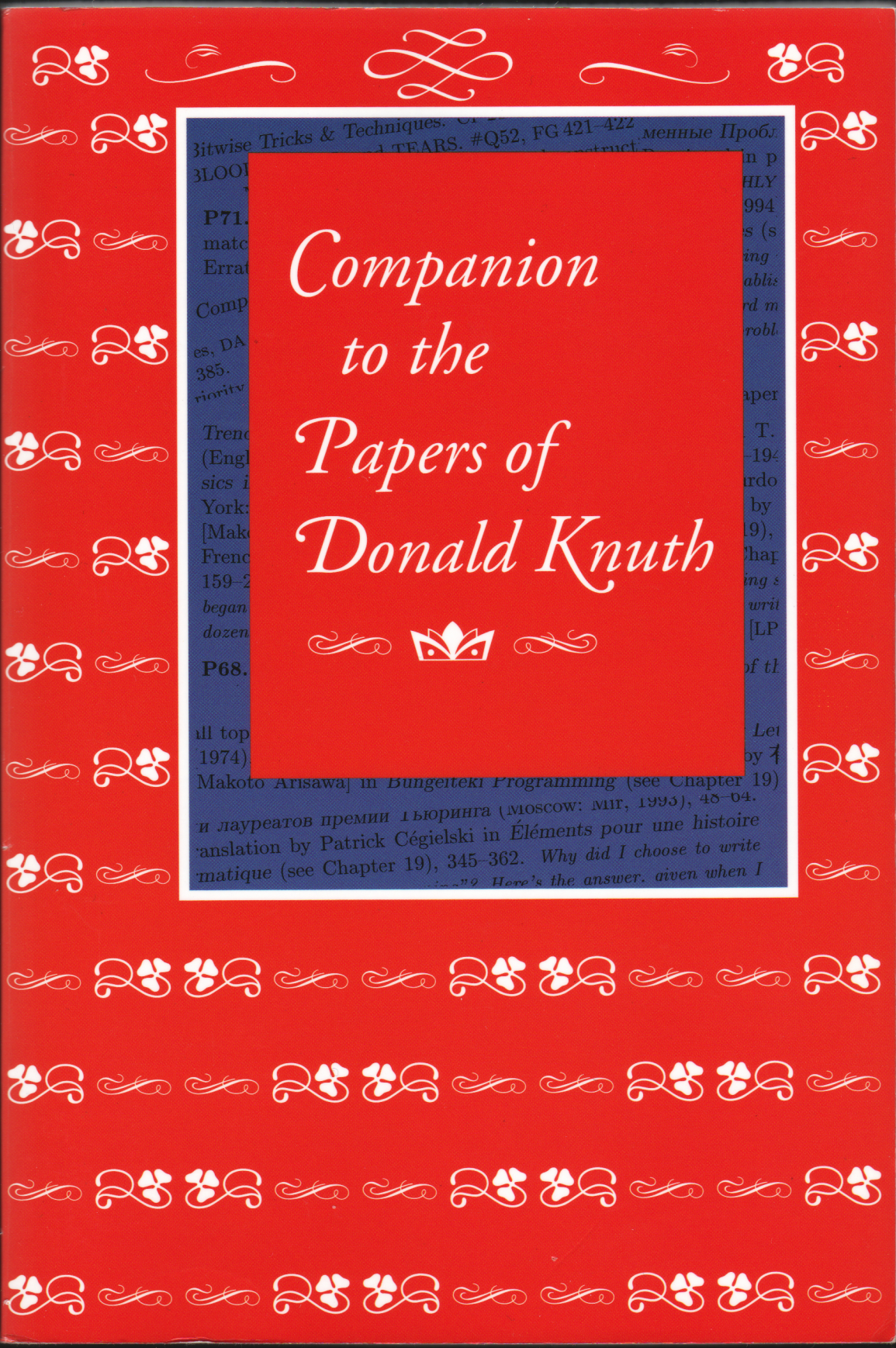


FIGURE 1. The evolution of complex components. Each configuration $[r_1, r_2, \ldots, r_q]$ stands for a graph or multigraph with r_1 bicyclic components, r_2 tricyclic components, $\ldots$, r_q $(q+1)$-cyclic components. As a graph evolves, its excess $r_1 + 2r_2 + 3r_3 + \cdots$ increases in unit steps, and the configurations follow a path from left to right in this partial ordering.



Selected Papers on Fun & Games

Donald E. Knuth


Companion
to the
Papers of
Donald Knuth


10
CRM Proceedings/Notes
Stable Marriage and Its Relation to Other Combinatorial Problems • Knuth
AMS
SEGUNDA EDIÇÃO
MATEMÁTICA CONCRETA
FUNDAMENTOS PARA A CIÊNCIA DA COMPUTAÇÃO
RONALD L. GRAHAM
DONALD E. KNUTH
OREN PATASHNIK
LTC
MATEMÁTICA CONCRETA
29
NÚMEROS SURREAIS
DONALD E. KNUTH
KNUTH
I
ALGORITMOS FUNDAMENTALES
editorial reverté, s.a.
KNUTH
III
CLASIFICACIÓN Y BÚSQUEDA
editorial reverté, s.a.
D. E. KNUTH
editorial reverté, s.a.
2e ÉDITION
MATHÉMATIQUES CONCRÈTES
R. Graham, D. Knuth et O. Patashnik
INTERNATIONAL THOMSON PUBLISHING
Le TEXbook
Donald E. Knuth
vuibert
KNUTH
Éléments pour une histoire de l'informatique
CSLI
SMF
KNUTH
Algorithmes
CSLI
SMF
KNUTH
Arithmetik
D. E. Knuth
Insel der Zahlen
GRAHAM KNUTH PATASHNIK
MATEMATICA DISCRETA
HOEPLI
ΔΕΥΤΕΡΗ ΑΜΕΡΙΚΑΝΙΚΗ ΕΚΔΟΣΗ
ΣΥΝΚΡΙΤΑ ΜΑΘΗΜΑΤΙΚΑ
ΜΙΑ ΘΕΜΕΛΙΩΣΗ ΓΙΑ ΤΗΝ ΕΠΙΣΤΗΜΗ ΤΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
GRAHAM
KNUTH
PATASHNIK
ΚΛΕΙΔΑΡΙΘΜΟΣ


Donald E. Knuth
Umění programování
1. díl Základní algoritmy
ADDISON WESLEY
CPRESS
Donald E. Knuth
Umění programování
2. díl Seminumerické algoritmy
ADDISON WESLEY
CPRESS
KNUTH
A SZÁMÍTÓGÉP-PROGRAMOZÁS MŰVÉSZETE
1
ALAPVETŐ ALGORITMUSOK
BUDAPEST
KNUTH
A SZÁMÍTÓGÉP-PROGRAMOZÁS MŰVÉSZETE
2
SZEMINUMERIKUS ALGORITMUSOK
BUDAPEST
KNUTH
A SZÁMÍTÓGÉP-PROGRAMOZÁS MŰVÉSZETE
3
KERESÉS ÉS RENDEZÉS
BUDAPEST
D. E. Knuth: Számok valóson innen és túl
RONALD L. GRAHAM / DONALD E. KNUTH / OREN PATASHNIK
KONKRÉT MATEMATIKA
MMIX. RISC számítógép az új évezrednek
1 1
DONALD E. KNUTH
A számítógép-programozás művészete
4 2
DONALD E. KNUTH
Kombinációk és partíciók előállítása
4 3
DONALD E. KNUTH
A számítógép-programozás művészete
4 4
DONALD E. KNUTH
Donald E. Knuth
TEX. Przewodnik użytkownika
KI
WNT
Donald E. Knuth
Sztuka programowania
Tom 1
KI
WNT
Donald E. Knuth
Sztuka programowania
Tom 2
KI
WNT
Donald E. Knuth
Sztuka programowania
Tom 3
KI
WNT

Donald E. Knuth
Sztuka programowania
Tom 1, zeszyt 1
KI
Donald E. Knuth
Sztuka programowania
Tom 4, zeszyt 2
KI
Matematyka konkretna
Основные алгоритмы
Искусство программирования для ЭВМ
Д. Кнут
Получисленные алгоритмы
Искусство программирования для ЭВМ
Д. Кнут
2
Сортировка и поиск
Искусство программирования для ЭВМ
Д. Кнут
3
Исправленное и дополненное издание
Кнут
Искусство программирования
Сортировка и поиск
Том 3
Второе издание
Addison-Wesley
Вильямс
Исправленное и дополненное издание
Кнут
Искусство программирования
Основные алгоритмы
Том 1
Третье издание
Addison-Wesley
Вильямс
Исправленное и дополненное издание
Кнут
Искусство программирования
Получисленные алгоритмы
Том 2
Третье издание
Addison-Wesley
Вильямс
Новинка
Кнут
Искусство программирования
Том 1 Выпуск 1
Т 1 В 1
AW
Новинка
Кнут
Искусство программирования
Том 4 Выпуск 2
Т 4 В 2
AW
Новинка
Кнут
Искусство программирования
Том 4 Выпуск 3
Т 4 В 3
AW
Новинка
Кнут
Искусство программирования
Том 4 Выпуск 4
Т 4 В 4
AW
Дональд Кнут
Сюрреальные числа
Конкретная математика
Р. Грэхем
Д. Кнут
О. Паташник
Д. Кнут
Все про METAFONT
Издание 2000 года
Вильямс
Addison Wesley

Д. Кнут
ВСЕ ПРО TEX
ИЗДАНИЕ 2000 ГОДА
ВИЛЬЯМС
ADDISON WESLEY
Все про TEX
КНУТ
AO RDTEX
Д. Э. Кнут
КОМПЬЮТЕРНАЯ ТИПОГРАФИЯ
Tratat de programarea calculatoarelor
Knuth D.E.
Tratat de programare a calculatoarelor
Algoritmi seminumerici
D. E. Knuth
Tratat de programare a calculatoarelor
Sortare și căutare
D. E. Knuth
KNUTH
Arta programării calculatoarelor
ALGORITMI FUNDAMENTALI
VOLUMUL 1
Teora
KNUTH
Arta programării calculatoarelor
ALGORITMI SEMINUMERICI
VOLUMUL 2
Teora
KNUTH
Arta programării calculatoarelor
SORTARE ȘI CĂUTARE
VOLUMUL 3
Teora
KNUTH
Arta programării calculatoarelor
Serie nouă
Teora
KNUTH
Arta programării calculatoarelor
Serie nouă
Teora
计算机程序设计技巧
国防工业出版社
计算机程序设计技巧
国防工业出版社
计算机程序设计技巧
第三卷
排序和查找
国防工业出版社
经典计算机科学著作最新修订版
计算机程序设计艺术
第1卷 基本算法
（第3版）
国防工业出版社
National Defence Industry Press
http://www.ndip.com.cn
Addison Wesley
经典计算机科学著作最新修订版
计算机程序设计艺术
第2卷 半数值算法
（第3版）
国防工业出版社
National Defence Industry Press
http://www.ndip.com.cn
Addison Wesley

经典计算机科学著作最新修订版
计算机程序设计艺术
第3卷 排序与查找
(第2版)
国防工业出版社
National Defence Industry Press
http://www.ndip.com.cn
Addison Wesley
HZ BOOKS 华章科技
PEARSON
计算机程序设计艺术
第4卷 第0册
(双语版)
(美) Donald E. Knuth 著
机械工业出版社
China Machine Press
博文视点
研究之美
[美] D. E. KNUTH 著
高博 译
電子工業出版社
具體數學
陳衍文 譯
儒林 SIM-452N
ADDISON WESLEY
具体数学 计算机科学基础
西安电子科技大学出版社
具體數學
者著原 Graham Knuth Patashnrk
賴飛羆 譯
東華書局印行
TURING 图灵教育
PEARSON
具体数学
计算机科学基础 (第2版)
[美] Ronald L. Graham Donald E. Knuth Oren Patashnik 著
人民邮电出版社
개정 3판
The Art of Computer Programming
기초 알고리즘
1
도널드 커누스 저 류광 역
Addison Wesley
한빛미디어
개정 3판
The Art of Computer Programming
준수치적 알고리즘
2
도널드 커누스 저 류광 역
Addison Wesley
한빛미디어
개정 2판
The Art of Computer Programming
정렬과 검색
3
도널드 커누스 저 류광 역
Addison Wesley
한빛미디어
KNUTH
1
基本算法／基礎概念
広瀬 健 訳
サイエンス社
KNUTH
2
基本算法／情報構造
米田信夫 筧 捷彦 共訳
サイエンス社
KNUTH
3
準数値算法／乱数
渋谷政昭 訳
サイエンス社
KNUTH
4
準数値算法／算術演算
中川圭介 訳
サイエンス社
1
The Art of Computer Programming
Fundamental Algorithms Third Edition
【日本語版】
アルゴリズムのバイブル
ASCII

Fascicle 1
1
The Art of Computer Programming
MMIX – A RISC Computer for the New Millennium
【日本語版】
アルゴリズムのバイブル
ASCII
2
The Art of Computer Programming
Seminumerical Algorithms Third Edition
【日本語版】
アルゴリズムのバイブルII
ASCII
3
The Art of Computer Programming
Sorting and Searching Second Edition
【日本語版】
アルゴリズムのバイブルIII
ASCII
F-0
4
The Art of Computer Programming
Volume 4 Fascicle 0
【日本語版】
アルゴリズムのバイブルIV
ASCII
F-1
4
The Art of Computer Programming
Volume 4 Fascicle 1
【日本語版】
アルゴリズムのバイブルIV
ASCII
Fascicle 2
4
The Art of Computer Programming
Generating All Tuples and Permutations
【日本語版】
アルゴリズムのバイブルIV
ASCII
F-3
4
The Art of Computer Programming
Volume 4 Fascicle 3
【日本語版】
アルゴリズムのバイブル IV
ASCII
F-4
4
The Art of Computer Programming
Volume 4 Fascicle 4
【日本語版】
アルゴリズムのバイブルIV
ASCII
数学小説
超現実数
D・E・クヌース
好田順治 訳
海鳴社
SURREAL NUMBERS
HOW TWO EX-STUDENTS TURNED ON TO PURE MATHEMATICS AND FOUND TOTAL HAPPINESS
至福の超現実数
純粋数学に魅せられた男と女の物語
D・E・クヌース 松浦俊輔 訳
数をめぐる尽きせぬ対話
柏書房
クヌース先生のドキュメント纂法
D.E. Knuth
T. Larrabee
P.M. Roberts
著
有澤誠 訳
クヌース先生のドキュメント纂法
D.E. Knuth
T. Larrabee
P.M. Roberts
著
有澤誠 訳
«Literate Programming»
文芸的プログラミング
Donald E. Knuth 著
有澤誠 訳
ASCII
SOFTWARE SCIENCE
クヌース先生のプログラム論
有澤誠 編
D.E.Knuth著作集
The TEXbook
TEXブック
コンピュータによる組版システム
Donald E. Knuth 著
監修 斎藤信男
翻訳 鷺谷好輝
ASCII
電子出版シリーズ
METAFONTブック
タイポグラファのためのプログラミング言語
Donald E. Knuth 著
鷺谷好輝 訳
ASCII
電子出版シリーズ
コンピュータの数学
CONCRETE MATHEMATICS
GRAHAM
KNUTH
PATASHNIK
著
有澤誠
安村通晃
萩野達也
石畑清
訳
共立出版
クヌース
コンピュータ科学者がめったに語ら
MITで神に会う!
SiB access
MMIXware
第三千年紀のためのRISCコンピュータ
ドナルド・E・クヌース 著
滝沢徹 訳
SiB access

Chapter 6

Theory and Practice and Fun

[Acceptance speech after receiving the 2010 Frontiers of Knowledge Award in Information and Communication Technologies from the BBVA Foundation, Madrid, 15 June 2011.]

Members of the Presidency, authorities, distinguished guests:

Of course I'm deeply honored to have been selected for this award, and deeply thankful to the distinguished computer scientists who have commented so favorably on my work. I'm also delighted to be receiving the award in Spain, since I've been a long-time fan of the concept of *siesta* ... and I love the Spanish language, especially the word *mañana*.

Occasionally in past years I've been asked to reflect on my life's work taken as a whole, and my response has always been to regard it as a mixture of two main themes: "theory" and "practice." I've devoted considerable time to humanity's ongoing quest for rigorous mathematical underpinnings, by which the fundamental ideas of computer science can be better understood and advanced. And I've also devoted roughly the same amount of time to the practical application of those theories, by writing computer programs that many people have found useful in their daily work.

The best theories are inspired by natural problems that arise in practice. The best practice is informed by theoretical results that quantify how much can be achieved by various methods. So there's a wonderful positive feedback loop, with theory grounded in practice and practice optimized by theory.

However, when I learned about the BBVA Foundation's award and was asked to prepare these brief remarks, it struck me that my life's work has in fact been characterized also by an important third dimension, namely "fun." My story is really best understood as a blend of theory, practice, and fun.

In fact my theoretical work has been driven by intellectual curiosity, by a compulsion to answer intriguing questions that seem to have begged






ΠΑΚΤΙΚΗ

DONALD E. KNUTH

3:16

BIBLE TEXTS ILLUMINATED

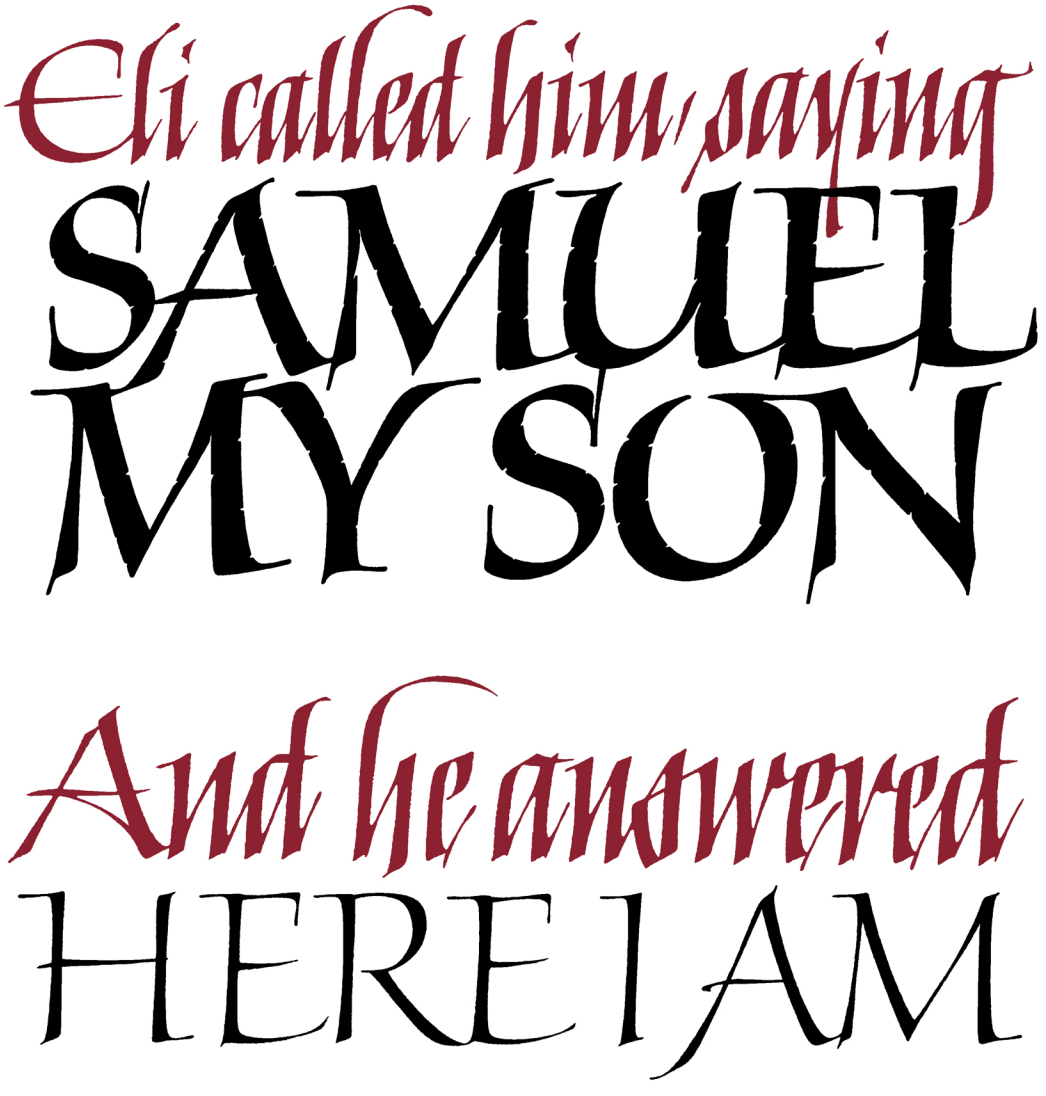
THE FIRST BOOK OF SAMUEL is the first half of what was originally a single, undivided work. It begins with the story of Israel's last two judges, Eli (4:18) and Samuel (7:15); it concludes with the story of Israel's first two kings, Saul (11:15) and David (16:1). Thus it tells about a significant turning point in Israel's history, when a loose federation of tribes became welded together into a single nation. This portion of the original Book of Samuel ends with an account of Saul's death and burial (31:6, 13).

The establishment of a monarchy was, of course, a highly controversial change in Israel's power structure and life style. Both sides of the debate are represented here. King Saul was not a smashing success, but David promised to be better. In retrospect, we can see that it was probably necessary for the Israelites to unify at this time, or they would have found it impossible to survive in the face of increased aggression from the Philistines, who had previously been content to live on the coast. The Israelites' forthcoming experiences as members of an affluent nation were destined to teach them important lessons about the dangers of material power and wealth; they gradually learned what sort of kingdom God really intended them to have.

This book can conveniently be regarded as a study of interactions between its leading characters. Chapters 1–7 are about Eli and Samuel; Chapters 8–15 are about Samuel and Saul; Chapters 16–31 are about Saul and David. The dominant personality is that of Samuel, a peaceful man who is especially influential as a spiritual leader; he points out that obedience to God is much more important than religious rituals (15:22). Later generations will remember Samuel as the first of the prophets, not only as the last of the judges (3:20; Acts 3:24, 13:20).

Like the other historical sections of the Old Testament, this book consists of a variety of heterogeneous accounts that were put together by later editors who have tried to include all the data at hand, even if the facts do not always dovetail perfectly. We find here a lively picture of Israel's early religious practices. The stories of Samuel throw considerable light on the pioneering activities of prophets and priests as well as kings.

The Hebrew text of Samuel has the dubious distinction of being the least well-preserved of all the Old Testament sources. For some reason, the ancient scribes who copied this material made more errors than usual, and the old manuscripts show considerable variation in small details. Some of the Dead Sea Scrolls have helped to resolve many of the textual riddles.



1 Samuel 3 : 16

Julian Waters

The call of Samuel differs from the other Bible stories about God's first confrontations with prophets, primarily because of Samuel's age: Samuel is probably very young at the time of these events. His mother has dedicated his life to God's service (1:28), and she has left him in the care of Eli, the high priest (2:11) and judge (4:18).

Americans do not pronounce the names Samuel and Eli as Hebrew speakers do. We say *SAM-you-ell* and *EE-lie*, but *shmoo-AIL* and *'ay-LEE* would be more authentic. (Samuel has the same Hebrew name as the Shemuel of Numbers 34:20.) No great harm is done when we pronounce things in our own way, but it is wise to remember that English might mislead us. We might think, for example, that Eli's name is closely related to that of Elijah, because of traditional English spellings; but in fact there is no connection between those names in Hebrew.

1 SAMUEL 3:16

Eli called him, saying, "Samuel, my son." And he answered, "Here I am."

Samuel is called by God in verses 4, 6, and 8; and he answers "Here I am" to Eli each time, because he doesn't recognize God's voice. But the call to Samuel in verse 16 is genuinely from Eli, and the circumstances are completely different this time when Samuel answers "Here I am" once again. We can learn several valuable things from this story, so it will be a good idea to look at it closely.

Verse 1 tells us that visions of God were infrequent in those days; thus it was a special honor for Samuel to receive a divine message. Eli was an old man, nearly blind, who was resting in his normal place (verse 2); this may have been near the door of God's sanctuary (1:9). The main thing that held the tribes of Israel together in those days was the existence of this sanctuary, which was then in the city of Shiloh, about 20 miles north of Jerusalem (Joshua 18:1; Judges 18:31). It was nighttime, because the lamp had been kindled (verse 3) as it was each night (Exodus 27:21; Leviticus 24:3). Samuel was lying by the Ark of God (verse 3), also called the Ark of the Covenant (4:4); this was a special portable shrine, symbolic of God's presence, which was kept in the inner sanctuary (Exodus 25:22, 26:33).

After three false starts, Eli realizes that Samuel is being addressed by God, so he advises Samuel to say "Speak, your servant is listening" (verse 9). This works (verse 10), and a channel of communication is opened. We learn from the story so far that we should be alert for God's messages to us (Psalm 85:8). Do we recognize that voice when it comes?

God says in verse 11 that he will be sending a terrible punishment to Israel in the years to come, something so terrible that it will "make everybody's ears tingle." This presumably refers to the capture of the Ark by the Philistines, as described in the next several chapters. The religious life of Israel has reached a truly abominable state under the corrupt leadership of Eli's sons (2:17, 22), and Eli's attempt at reprimanding them has had no effect (2:24–25). Their rejection of God is unforgivable (verse 14).

God's first message to Samuel was certainly not calculated to produce peace of mind. Right from the start Samuel has to bear the prophet's cross, the painful duty of telling unwelcome truths to people he loves (see Jeremiah 15:10). He has just learned of God's anger—an anger so great that it will destroy his present home. Therefore he is afraid to tell Eli what has happened; he lies in bed, probably unable to sleep (verse 15).

However, when morning comes, Samuel opens the sanctuary doors, knowing that Eli will be nearby. It is Samuel's duty as a Korahite to tend these doors (1 Chronicles 6:33–37, 9:19). Throughout this story, Samuel performs his tasks faithfully.

Now comes verse 16, where we started. Eli calls out, "Samuel—my son!" He knows that his real sons have betrayed God's trust; his adopted son, Samuel, must be the one who will carry God's work forward. Saint Paul, similarly, calls Timothy his "true son in the faith" (1 Timothy 1:2).

Eli surely anticipates the worst, because he has already been warned (2:33). He is a man of great power, as head of both church and state in Israel; therefore we might expect him to be jealous and to give Samuel the cold shoulder. Why should God bypass the high priest for a child? But he continues to act in a fatherly way, and he presses Samuel to tell him everything. He invokes a solemn oath in verse 17, perhaps making a hand gesture something like slitting his throat when he says the words; this sort of oath was common in those days (see 14:44, 20:13, 25:22; Ruth 1:17).

Samuel holds nothing back (verse 18), and this leads to the climax of the story: Eli calmly accepts God's judgment, terrible as it is, without whining or complaining. "He is Jehovah; let him do what seems best." The same lesson is taught in many other places in the Bible when great men submit their own egos to God's divine ordinances. We learn this, for example, from Aaron (Leviticus 10:3); from David (2 Samuel 15:26; Psalm 39:9); from Hezekiah (2 Kings 20:19); and from Jesus (Luke 22:42).

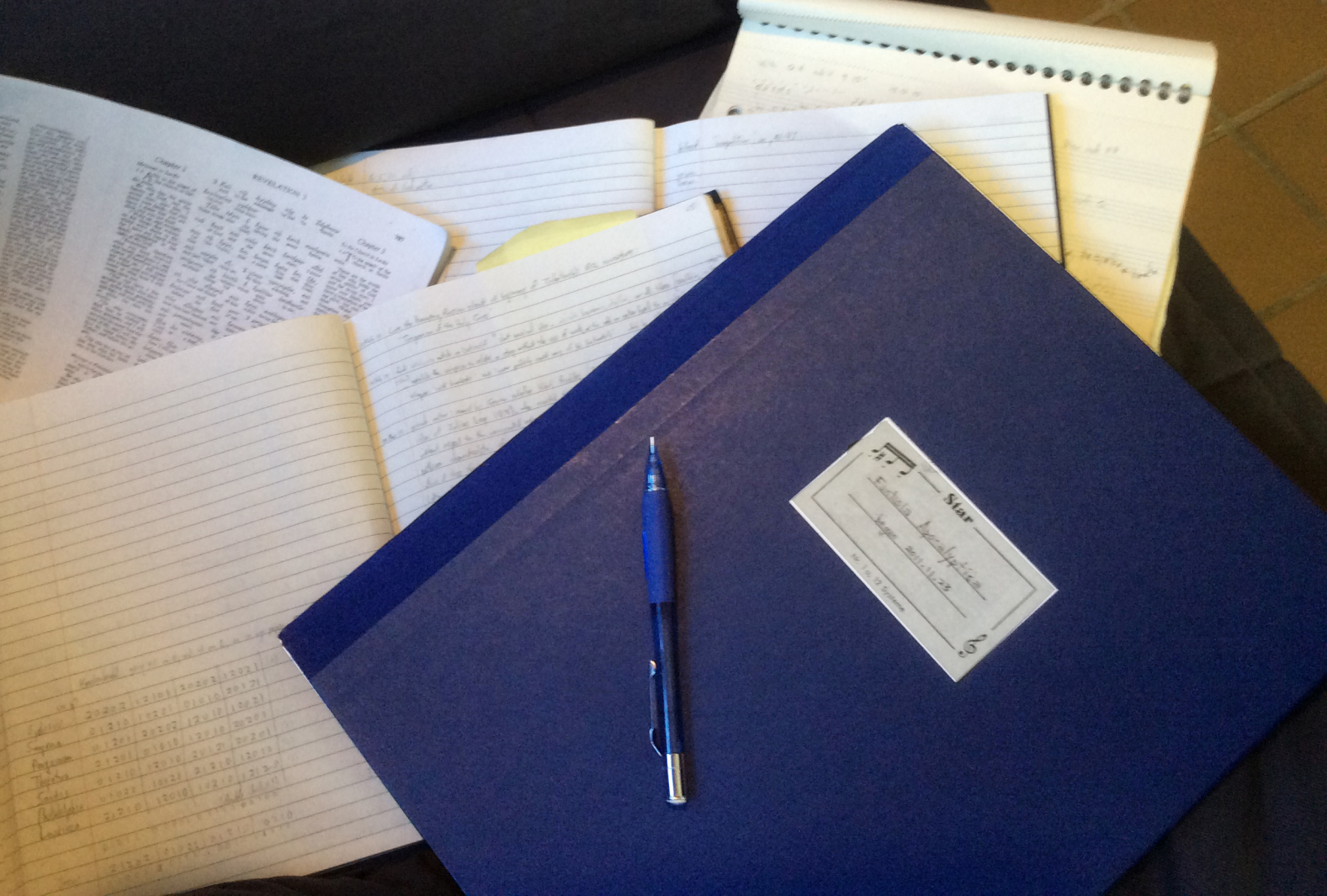


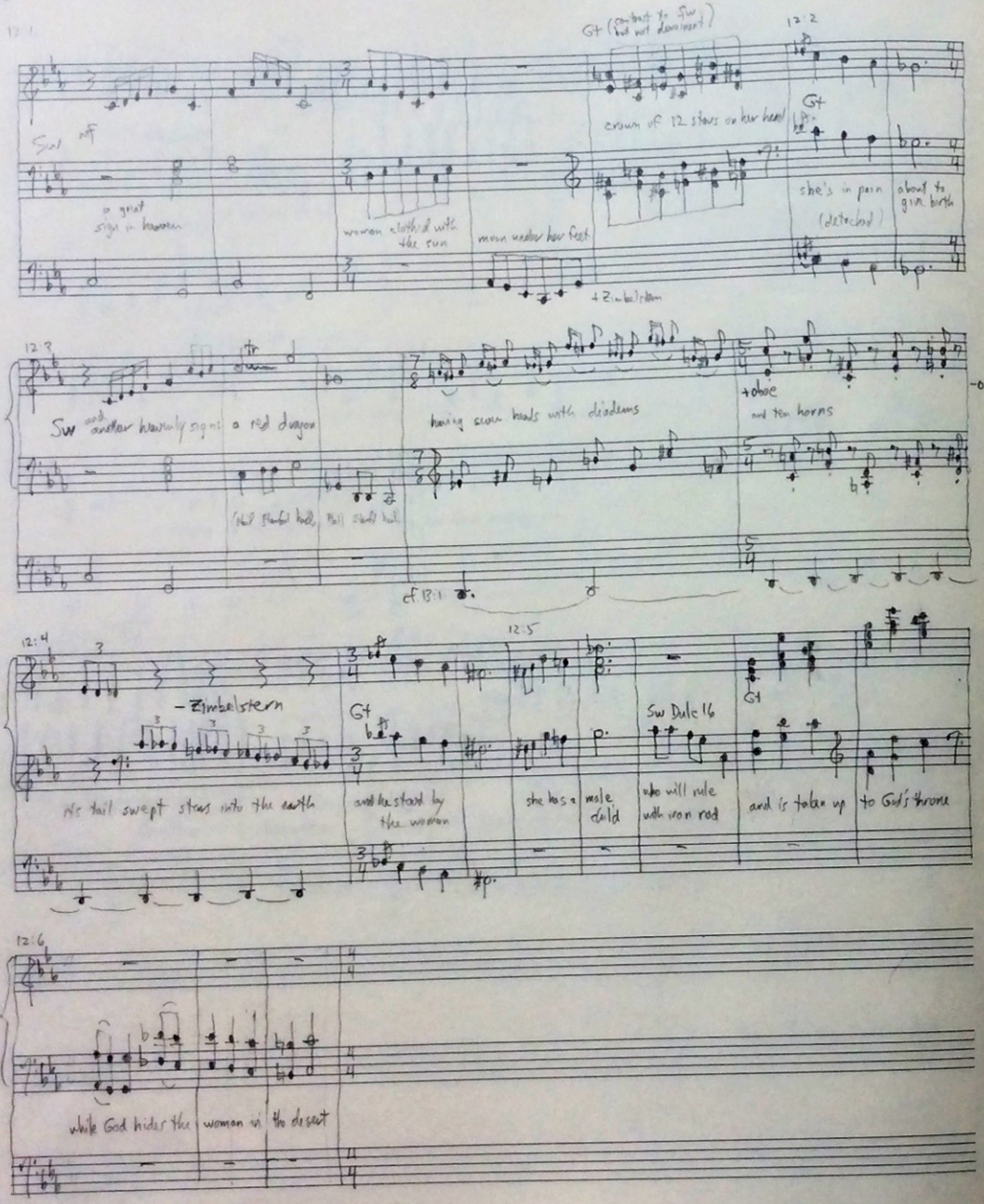
Donald Knuth playing the piano, Milwaukee, WI; May 1940



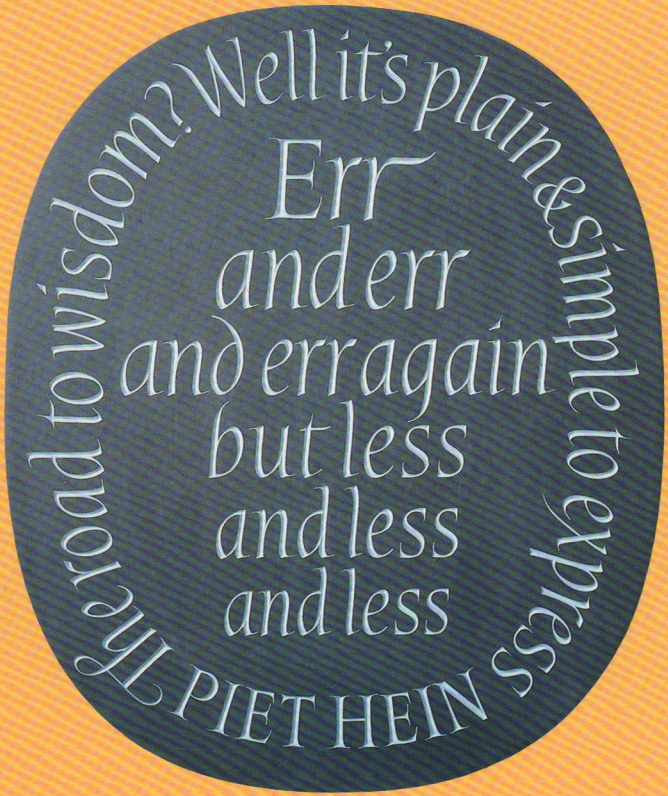





Star



STAR® Nr. 1, 12 Systeme®


The road to wisdom? Well it's plain & simple to express
Err
and err
and err again
but less
and less
and less
PIET HEIN