An appreciation: *The Art of Computer Programming, Volume 4A*

David Walden

Donald E. Knuth, *The Art of Computer Programming, Volume 4A: Combinatorial Algorithms, Part 1.* Addison-Wesley, Boston. Jan. 2011. 912 pp. Hardcover, US$74.99. ISBN 0-201-03804-8.

A book of algorithms relating to computer programming and their analysis (and about problem solving more generally) would not normally be reviewed in this journal about typesetting and fonts. However, TEX, TUG, and *TUGboat* would not exist without the problem Knuth faced in 1976 with the the typesetting of the second edition of Volume 2 of *The Art of Computer Programming* (*TAOCP*); and thus *TAOCP* is a key part of the history of the TEX world. Many readers of this journal already know this story (told in chapter 1 of Knuth's book *Digital Typography*).[1] Addison-Wesley had gotten rid its Monotype technology in 1963 and could not reproduce with the photo-optical machines of the time the high quality typesetting of the original printings Volumes 1–3 (and new printings of Volumes 1 and 3 done with Monotype machines still found in Europe). Consequently, in 1977 Knuth began developing a new typesetting system to restore the high quality typesetting of books, in particular *TAOCP*. Eventually TEX was available and became popular with various groups of users; and the TEX Users Group and *TUGboat* came into being.

I bought Volumes 1, 2, and 3 of this series immediately after publication of each book in 1968, 1969, and 1973 and used them frequently in my profession as a computer programmer. I also bought new editions of these books as they came out over the years, keeping my complete set of *TAOCP* up to date. Thus, to maintain my complete set of *TAOCP*, I bought Volume 4A immediately upon its publication and have been dipping into it to get an overall sense of it. (As I suspect is the case with many other readers of this series, I have never read a volume completely through, but rather skimmed each book enough to know what was in it and then studied particular sections more deeply as needed for the project on which I was working, or when I just wanted to have some fun reading about algorithms.)

Over the years since the original editions of volumes 1–3 were published, Knuth's original plan for a 7-volume, 1–chapter series has gradually evolved as some the topics of his originally planned chapters have been covered in depth by other books and as the topics covered by some of the chapters expanded dramatically (perhaps partially as a result of Knuth's example of rigorous, comprehensive books describing computer algorithms). Today Knuth hopes to produce five volumes, of which the current volume 4 will have at least three parts (books): `http://www-cs-faculty.stanford.edu/~uno/taocp.html`

Part 1 (that is, book 1) of Volume 4 (the overall volume is on combinatorial algorithms) covers an Introduction, Zeros and Ones (with four subsections) and Generating All Possibilities (with one subsection containing seven subsubsections). Curiously, the second and third subsections on Generating All Possibilities are not due until Volume 4B. Perhaps at 912 pages (and after publication of the groups of pages from the book over the past half dozen or more years as a succession of five fascicles), Knuth or the publisher decided that Volume 4A was already long enough.

As with the previous volumes of *TAOCP*, this book is substantially about the analysis of the algorithms presented and not just a cookbook of algorithms. A reader can either just find what Knuth says is the best method and use it, or can learn why it is a good method, why other methods are not so good, and how to do the math to analyze the performance of one's own situations where the algorithms might be used. The book also includes Knuth's usual sets of exercises and hundreds of pages of answers to exercises.

Of the parts of Volume 4A I have touched on so far, I greatly enjoyed the discussion of Latin and Greek squares (the clearest I have ever read), and I know I am going to enjoy reading more of the discussion on bitwise tricks and techniques, a topic that has always fascinated me. I also have looked at some of the resources on Knuth's web site augmenting discussions in the book (and the reader's own use of the methods Knuth describes and analyzes). I also enjoy skimming pages of Knuth's *TAOCP*, skipping the real math and reading bits of math-and-algorithm history. Section 7.2.1.7, History and further references, looks like it will be particularly fun reading. I never try to work any *TAOCP* exercises but rather will dip straight into the comprehensive answer sections to find additional information I need that is not covered in the main text.

Not being a mathematician myself, I sought out a comment on the book from mathematician (and puzzle master) Bill Gosper[2] (who has four entries in

[1] CSLI Publications, Stanford, CA, 1999

[2] `http://en.wikipedia.org/wiki/Bill_Gosper`

Comment from Bill Gosper

I am delighted to report that Knuth is still his usual precise, profound, and playful self.

The book is surprisingly therapeutic — it will help you lose any guilt you may feel over designing and working puzzles.

On page 172 Knuth says: "For example, after Martin Gardner introduced John Conway's game of Life to the world in 1970, more computer time was probably devoted to studying its implications than to any other computational task during the next several years — although the people paying the computer bills were rarely told!"

However, the above follows his inflammatory remark on page 2: "On the other hand, the exact value of L[angford arrangement]$_{100}$ will probably never be known, even as computers become faster and faster."

Has Knuth any idea how many computational resources will now be expended trying to prove him wrong?

On page 2: "In Section 7.2.2.1 we shall study an algorithm called 'dancing links,' which is a convenient way to generate all solutions to such problems."

And on page 464: A technique called "dancing links," which we will discuss extensively in Section 7.2.2.1, is used here to remove and restore items from and to doubly linked lists.

At last, my chance to hear it from the Master! Eagerly flipping forward, . . . , 7.2.1.6, 7.2.1.7, Answers to Exercises. ARGHH! To Be Continued!

And to think I had been salivating over page viii: "(See the discussion of NP-completeness in Section 7.9.)" !

The Table of Contents looks positively meager. How could this require 883 pages? Clue: The Index takes 50 pages. Open to one page at random. Can you plow through it in an hour? A day? This is no cookbook. Don't open it unless you plan to *learn* something. 34.4 percent of the book is Answers to Exercises.

PS, Don't miss Knuth's brilliant new twist on "This page intentionally left blank."

the Volume 4A index). Bill's reply (email of June 3, 2011) is in the sidebar.

Volume 4A looks like the previous volumes (in their latest editions) — the same design and great care with details (the Knuthian way). In my memory, some details have evolved since the first edition of Volume 1. For instance, with each new volume and each new edition (maybe even with new printings), including the middle names of cited people and correct presentation of their names in their own language have become ever more complete.

Knuth's editor at A-W, Peter Gordon, has stated that the A-W production staff sees the end product of Knuth's work; Knuth supplies PostScript files to A-W, which the A-W printer converts to PDFs. The colophon of Volume 4A says, "This book was produced on an HP Compaq 2510p using Computer Modern typefaces, using TeX and METAFONT software as described in the author's books *Computers & Typesetting* (Reading, Mass.: Addison-Wesley, 1986), Volumes A–E. The illustrations were produced with John Hobby's METAPOST system." A close look by Karl Berry at a PDF page from the book suggested that Knuth is using `tex|dvips` and his original bitmapped CM fonts, not the Type 1 fonts that are the default in current TeX distributions. (While providing the rest of us with a system that has been extended in many ways, Knuth apparently sticks with the original system he created to produce a beautiful edition of *TAOCP*, using his tool to control every pixel on the page.)

In the world of computer historians (i.e., often people who have not been computer professionals themselves), there was some interesting commentary immediately after Volume 4A of *TAOCP* was published. In essence the question (maybe tongue in cheek) was what took Knuth so long, given how profoundly volumes 1–3 impacted the field of computing — why did he delay this most important work to do other things judged less important by computer historians.

In my view, the historians may over-estimate the impact of *TAOCP* on the field of computer programming. Most programmers don't use the books, and many people don't think highly of the books as potential textbooks for typical courses for computer programmers. (Volumes 1–3 clearly did have deep impact in their early comprehensive and rigorous coverage of a range of parts of what was becoming the discipline of computer science.)

The historians may also under-estimate the importance of Knuth's other work. In my view, Knuth in his field is like Picasso in his. He has had multiple simultaneous and serial careers, any of which would be more than the lifetime achievement of most people. Writing *TAOCP* is one of Knuth's most important achievements, but I don't think it is singularly important.

As I see it, the first three volumes of *TAOCP* revolutionized how to analyze algorithms for the purpose of accomplishing some task in a computer program. From these books, I learned new algorithms to use in my programming, and I learned about how to think better about methods I and my fellow programmers were already using (sorting, hashing, . . .). (By the way, I agree with Knuth's often criticized decision to write the books using assembly language for his hypothetical original and more modern machines rather than in a high-level language.)

Volume 4A is not such a revolution because Knuth no longer can give comprehensive coverage (and because he already showed us the path to rigorous analysis of algorithms in Volumes 1–3). As Knuth has explained, after volumes 1–3 of *TAOCP*, the field exploded, and he no longer could do what he set out to do. Nonetheless, Volume 4A is a further example of his stunning ability to understand vast amounts of material, choose interesting parts, and present them in a fascinating way.

As noted, Knuth also felt the need to work on digital typesetting so a revision of Volume 2 of *TAOCP* would not look bad to him. In so doing, he revolutionized digital typesetting and font design. This incidentally gave many mathematicians, physicists, economists, etc., a new way to do their writing and began what is probably the longest running open source software success story. Over the years the breadth of use of TEX et al. has continued to grow (critical editions of literature, non-Latin alphabet writing, etc.), even as commercial typesetting systems with their GUI interfaces have become the norm in the population at large (with these commercial systems gradually adopting most of the algorithms TEX had long ago). Some might argue that TEX and Knuth's investigations into digital typesetting and font design have had more impact on the world than Knuth's self-proclaimed masterwork, *TAOCP*.

It also may be that, after the first three volumes of *TAOCP*, Knuth had to regroup to ready himself for the next step in the series.

- He wrote two books organizing and extending the math approach to analysis of algorithm: one with Daniel Green [*Mathematics for the Analysis of Algorithms*, third edition, Birkhäuser, 1990], and one with Ronald Graham and Oren Patashnik [*Concrete Mathematics*, second edition, Addison-Wesley, 1994].
- He created a new, modern hypothetical machine to use for his examples and wrote the book about the machine [*MMIXware: A RISC Computer for the Third Millennium*, Springer-Verlag, 1999].
- He created the Stanford GraphBase system to help with combinatorics problems and wrote the book [*The Stanford GraphBase: A Platform for Combinatorial Computing*, ACM Press, 1994].
- Also in this period, he brought out the eight or so volumes of his collected papers, some of which could be a good text for a seminar in some of the topics in *TAOCP* as originally conceived which he is now never going to get to. He brought out the most recent of these volumes in late 2010.
- Of course, he also developed the concept of literate programming and the software to compile and document large systems such as TEX [*The CWEB System of Structured Documentation*, with Silvio Levy, Addison-Wesley, 1993].

(Knuth also did some research not related to computer science: for instance, a carefully created book relating to Bible study (see `http://www.tug.org/TUGboat/tb12-2/tb32reviews.pdf`); but who is to say that that Bible study was not also a useful preparatory step toward Volume 4 of *TAOCP*.)

Apparently finally Knuth was ready again to tackle Volume 4 which, after resigning as a Stanford professor to have more time, he has been doing for a number of years now (e.g., pushing out the fascicles). Despite all Knuth's contributions in a variety of areas, he still felt the need to finish what he calls "the interesting parts" of his original plan for *TAOCP*. The preface to Volume 4A now suggests (to me) that he may never get beyond vol 4B, 4C, ... (I don't know if he intended to say this — he does say he will surely never get to 4Z.)

My admiration for Knuth is unbounded. Volume 4A is another stunning example of Knuth's breadth, thoroughness, and desire to produce a beautiful book.

As a purely personal matter, I adopted the use of TEX-based systems when I made the decision in the 1990s to stop using word processing systems with graphical user interfaces. I chose TEX as my replacement word processing system because of my admiration for Knuth and the desire to use something he had created. I haven't been disappointed.

More generally, I stand by my comparison with Picasso. Knuth is as much an artist as he is a technician. He goes where his muse takes him and he does so with unmatched (for the computer field) skill, care, depth, breadth, and artistry. We in the TEX community remain major benefactors of Knuth's skill and artistry.

For someone like me who still feels a strong connection to the world of computer programming, there is another thing to marvel at regarding Knuth. He is perhaps unique as a person of his stature as a theoretician for how many lines of code he apparently still writes every day. For Knuth programming is an art he practices every day.

◇ David Walden
`http://www.walden-family.com/texland`