

Book Review

The Art of Computer Programming, Volume 4, Fascicle 4, by D.E. Knuth, Addison Wesley Professions, Boston, MA, USA, 2006. ISBN: 0321335708

Donald Knuth is well known as the author of a three volume encyclopedic reference text on algorithmic computer science [1–3]. *The Art of Computer Programming* volumes have been authoritative references since first published in the late 1960's. Knuth pays fanatic attention to bibliographic and technical details, paying a bounty of \$2.56 to the first finder of each error. The books are also famous for Knuth's exercises, which have challenged generations of students.

Knuth originally planned to complete the series with four more volumes, on combinatorial algorithms, syntactical algorithms, the theory of languages, and compilers. Unfortunately other projects, including the TeX text processing system, intervened. Knuth also wanted to revise the first three volumes to reflect progress made during the intervening thirty years. In the late 1990's Knuth completed revisions to the first three volumes of the series and began work on the fourth volume. As Professor Emeritus of The Art of Computer Programming at Stanford, Knuth recently made substantial progress on the fourth volume in the series. Volume 4, *Combinatorial Algorithms*, is now projected as three physical volumes, with volume 4A on enumeration and backtracking, 4B on graph and network algorithms, and 4C on combinatorial optimization and recursion. Drafts of the new material are being released in sections, called fascicles, of about 128 pages each.

Fascicle 1 describes MMIX, an updated version of MIX, the computer architecture and assembly language used by Knuth to describe low level algorithms [4]. The material on MMIX is actually planned for a revised version of volume 1, and Knuth makes little use of it in the other fascicles from volume 4 that were recently published. Thus fascicles 2, 3, and 4 can be read independently of [4].

Counting families of combinatorial objects such as permutations and combinations is an important topic in combinatorics but most presentations of enumerative combinatorics ignore the problem of systematically

listing the objects. Knuth is interested in algorithms for the systematic listing of combinatorial objects including n -tuples, permutations, combinations, partitions of integers, set partitions, and trees. In the second fascicle, Knuth considers the generation of n -tuples and permutations [5]. The third fascicle deals with generating combinations and integer and set partitions [6].

The fascicle under review, #4, completes the coverage of combinatorial generation in volume 4A with a section on the generation of trees. The section on generating all trees stands on its own and can be read without reference to the material in the previous fascicles. A second section gives a history of algorithms for generating combinatorial objects. This section includes many references to material in fascicles 2 and 3.

In Section 7.2.1.6, Knuth considers the problem of generating all trees on n nodes. A key idea is the correspondence between strings of properly nested parentheses and trees. The number of properly nested strings of parentheses of length $2n$ (and the number of trees on n nodes) is well known as the Catalan number,

$$C_n = \binom{2n}{n} - \binom{2n}{n-1}.$$

Although this solves the problem of counting the number of trees on n nodes, generating all of the trees in a systematic fashion is a different problem.

Knuth discusses algorithms for generating all trees on n nodes in lexicographic order, determining where a tree occurs in the lexicographic ordering (ranking), finding the k th tree in the sequence (unranking), and generation of random trees. In addition to algorithms based on the lexicographic ordering, algorithms are given for the generation of trees in gray code order, in which successive trees differ in only one position. Knuth also gives algorithms for the generation of all spanning trees on a given graph. Much of the material in this section is well known [7–9]. What makes this section particularly interesting is the collection of 124 exercises that complete the section. These range from straight forward exercises that can be solved in a few minutes to challenging problems that will require

considerable effort to solve. Each exercise is marked with a difficulty level from 00 to 50, where a level 00 problem should be immediately solvable and a level 50 problem is an unsolved problem whose solution would be publishable.

In Section 7.2.1.7, Knuth discusses the history of algorithms for generating combinatorial objects. The history of the subject goes back thousands of years. He gives a fascinating account of early ad hoc and systematic listings of combinatorial objects as diverse as the hexagrams of the *I Ching*, metrical feet in Greek poetry, and rhythmic patterns in music. The modern history of the subject starts in the 1950's with the advent of digital computers. As usual, Knuth provides a very extensive collection of references. This section includes a set of 32 exercises, but these are somewhat less interesting than the exercises in Section 7.2.1.6. Many of the exercises simply ask the reader to complete historical examples.

The fascicle concludes with solutions to the exercises and an extremely thorough index. Bibliographic references are scattered throughout the fascicle, so there is no conventional bibliography.

Although the author doesn't claim that this is a complete and final version of the work, the quality is very high. The material that has already appeared in the fascicles could be compiled into a very respectable book today. With the feedback that the author is getting from readers of the fascicles, there is no doubt that the final published version of this material will have very few errors. Fascicle 4 is recommended to readers with a particular interest in the generation of trees or the history of combinatorial generation. It will also interest many fans of Knuth's work who just can't wait for the complete volume 4.

References

- [1] Donald E. Knuth, *Fundamental Algorithms*, Addison-Wesley, Reading, Massachusetts, third edition, 1997.
- [2] Donald E. Knuth, *Seminumerical Algorithms*, Addison-Wesley, Reading, Massachusetts, third edition, 1997.
- [3] Donald E. Knuth, *Sorting and Searching*, Addison-Wesley, Reading, Massachusetts, second edition, 1998.
- [4] Donald E. Knuth, *The Art of Computer Programming*, Volume 1, Fascicle 1: MMIX – A RISC Computer for the New Millennium. Addison-Wesley Professional, Reading, Massachusetts, 2005.
- [5] Donald E. Knuth, *The Art of Computer Programming*, Volume 4, Fascicle 2: Generating All Tuples and Permutations. Addison-Wesley Professional, Reading, Massachusetts, 2005.
- [6] Donald E. Knuth, *The Art of Computer Programming*, Volume 4, Fascicle 3: Generating All Combinations and Partitions. Addison-Wesley Professional, Reading, Massachusetts, 2005.
- [7] Donald L. Kreher and Douglas R. Stinson, *Combinatorial Algorithms: Generation, Enumeration, and Search*, CRC Press, Boca Raton, 1999.
- [8] Albert Nijenhuis and Herbert S. Wilf, *Combinatorial Algorithms*, Academic Press, San Diego, second edition, 1978.
- [9] Herbert S. Wilf, *Combinatorial Algorithms: An Update*, SIAM, Philadelphia, 1989.

Brian Borchers
Department of Mathematics
New Mexico Tech
Socorro, NM, USA