APPEARED IN BULLETIN OF THE
AMERICAN MATHEMATICAL SOCIETY
Volume 27, Number 2, October 1992, Pages 313-315

BOOK REVIEW

Evolution of random search trees, by Hosam M. Mahmoud. Wiley, New York, 1992, 324 pp., $54.95. ISBN 0-471-53228-2

In everyday life we usually store files in a filing cabinet in alphabetical order, which makes it easy for a person to locate an existing file or insert a new file. But in a computer, assigning storage locations in this contiguous linear way is less efficient, because (for instance) inserting a new file in the middle would require half the existing files to be moved. One alternative (among many) is to regard the storage locations as arranged in a binary tree. If files arrive in the order "PIG DOG SHEEP CAT DUCK COW RABBIT" we can store them successively, without moving previous files, as shown in the figure.

The rule for adding a new file, say FOX, is to compare with the root (PIG). Because FOX precedes PIG in alphabetical ordering we move to the left child (DOG) and compare FOX with DOG; because FOX does not precede DOG we move to its right child (DUCK), and after another comparison we finally store FOX as a right child of DUCK. Note that the same procedure enables us to locate an existing file. This data structure is the *binary search tree* In the 7-item example pictured, the average number of comparisons needed to locate a file is $(1+2+2+3+3+3+4)/7$. This number, and the shape of the tree, would typically be different with a different ordering of arrival of the files. But a natural probabilistic model is that all $n!$ possible orderings of n files are equally likely, and under this model one can study the expected average number c_n of comparisons needed to locate a file, and it turns out that

$$c_n \sim 2 \log n. \tag{1}$$

This result is classical (for computer science) and, indeed, tree-based searching structures occupy about eighty pages of the famous work of Knuth [1]. The present book is devoted to extensions of (1) in three directions.

(a) Replace "average number of comparisons" by some other measure of efficiency (e.g., "maximum number of comparisons"), and study its expectation.

(b) Replace the particular binary search tree with trees constructed via different rules (the book discusses m-ary trees ($m > 2$), quad trees, tries, and digital search trees).

(c) Refine results on expectations to give results on asymptotic distributions.

The book seems intended partly as a graduate text for computer science and partly as a record of recent research results and methods. It does a very good job

FIGURE 1

on basics: describing carefully the different mathematical models and their implementation as algorithms and giving a clear exposition of the simpler mathematical analysis. On the other hand, some of the sections dealing with recent results read more like research articles than a textbook. Mathematicians should be warned that little attempt is made to connect the rather narrow topic of the book to the larger world of computer science. For instance, Chapter 4 treats quad trees, a method of storing locations of points in n-space. This is a rather minor aspect of the large and thriving field of computational geometry (see, e.g., Preparata and Shamos [2]) devoted in part to methods of storing positions of objects in space in such a way that specified types of questions (convex hulls, nearest neighbors, line of sight) may be answered quickly. But uninitiated readers get no hint that this huge field exists.

Turning to questions of mathematical techniques for solving probability problems arising in computer science, one extreme is represented by what I call the "Knuth school," whose strategy is to transform immediately to analytic problems involving recurrences, generating functions, and their asymptotic analysis via the Darboux method and Mellin transforms and their inversion by contour integration. This school (see Vitter and Flajolet [3] for a recent survey) has in the past paid almost no attention to the work of mainstream mathematical probabilists over the last 40 years. Conversely, the mainstream has largely focused on continuous problems and ignored concrete discrete problems. Fortunately, this academic Cold War is also ending. Thus the present book, though its heart is still with the Knuth school, does contain uses of modern mainstream methods, such as the martingale analysis of internal path length of binary search tree and the branching random walk analysis of their height. There are certainly other mainstream ideas on the shelf ready to be used (e.g., the proofs of trie height asymptotics can surely be simplified by appealing to the Poisson limit theorem for U-statistics). Conversely, computer science examples such as the binary search tree and the Metropolis algorithm are more deserving of space in introductory stochastic processes texts than time-worn examples like "type 2 counters." More interaction between mathematical probabilists and those applying probability to discrete problems will benefit both groups.

REFERENCES

1. D. E. Knuth, *The art of computer programming*, vol. 3, Addison-Wesley, Reading, MA, 1973.
2. F. P. Preparata and M. I. Shamos, *Computational geometry*, Springer-Verlag, New York, 1985.
3. J. S. Vitter and P. Flajolet, *Analysis of algorithms and data structures,* Handbook of

Theoretical Computer Science, vol. A: Algorithms and Complexity, Ch. 9, North-Holland, Amsterdam, 1990, pp. 431–524.

David Aldous
University of California at Berkeley