student, it would not be, for its view of programming is too abstract and specialized.

PAUL ABRAHAMS

Courant Institute of Mathematical Sciences
New York University
New York, New York 10012

48 [12, 13.40].—E. S. PAGE & L. B. WILSON, *Information Representation and Manipulation in a Computer,* Cambridge Univ. Press, New York, N. Y., 1973, v + 244 pp., 23 cm. Price $5.95.

I was quite pleased to receive this book for review, since I have been searching for a long time for a satisfactory text for a course in data structures; this is the first text to fill the need. The classic text on the subject is Knuth's "Art of Computer Programming", but the relevant material is spread over two volumes. Furthermore, the explanations are on too high a level for an undergraduate course. Page and Wilson's book might well be entitled, "A Simplified Version of Knuth". Virtually all of their material, except for an early chapter on symbols and codes, is contained in Volumes I and III of Knuth, and their references to Knuth are plentiful.

Chapters 1 and 2 deal with symbols and their encodings and include a treatment of error detection and correction and Huffman coding. These topics seem somewhat remote from the rest of the book. Chapter 3 discusses internal representations of numbers. This would be an appropriate point for a discussion of character strings, but there is none, perhaps because Algol does not have any facilities for manipulating them. Chapter 4 treats arrays and methods of accessing them, and includes a good discussion of sparse arrays (arrays that are mostly zero). Chapter 5 discusses allocation methods for queues, stacks, and deques, considering both arrays and linked lists. There is also a more general discussion of linked lists here. Chapter 6 discusses trees and how to traverse and represent them. The discussion of conversion between binary and general trees seems overblown, however. Chapter 7 discusses searching, including binary and hashcoded searches, and Chapter 8 discusses sorting methods.

There are a number of disconcerting flaws in the book. Explanations are occasionally obscure (e.g., an application of generating functions on p. 158). The justifications given for circular lists (p. 86) are incorrect; a circular (singly-linked) list saves one pointer and nothing else, compared with an ordinary list with front and rear positions. Algol is used throughout as the programming language, and the limitations of Algol (no strings, no pointers) cause these programs, as well as the choice of methods, to be unnecessarily obscure. The book is well supplied with exercises, but many of these are taken from British university examinations and require nothing more than a regurgitation of material in the preceding chapter.

On the plus side, the book is free of major idiosyncracies (a frequent problem with computer science texts) and the explanations are generally clear. One can ask students to read the book without forcing them to get involved in extraneous auxiliary material. Despite its shortcomings, I recommend this book as a text for a data structures course.

Paul Abrahams