on an optimal path for which $f_s(m) \leqslant H(s, t)$. Contradiction since m would have been chosen. QED 5

Proof 3:

Suppose not. Let, on iteration i, $n \in \bar{S}$ and $m \in \bar{T}$ be the first pair of nodes for which holds that $\bar{H}$ must expand one of them while $\bar{H}_A$ will expand neither of them. At some iteration l and at all following iterations n and m will be both in $\bar{S}_A$ resp. $\bar{T}_A$ (because n and m were the first nodes not expanded by $\bar{H}_A$, their ancestors must be expanded at some moment).

Denote the f_s-value using $\bar{H}_A$ on iteration l by f_A^l. So we get:

$$\begin{aligned} f_A^l(n) &= g_s(n) + \min_{y \in \bar{T}} (\bar{H}_A(n, y) + g_t(y)) \\ &\leqslant g_s(n) + \bar{H}_A(n, m) + g_t(m) \\ &< g_s(n) + \bar{H}(n, m) + g_t(m) \\ &= f_s(n) \; (= f_t(m)) \end{aligned}$$

on iteration i of $\bar{H}$ and on iteration k where $\bar{H}_A$ halts, in say 1, with a solution path we get:

$$H(s, t) = f_A^k(1) \leqslant f_A^k(n) < f_s(n) \text{ (on iteration } i) \; .$$

But now we have a contradiction with lemma 5. QED 3

References

NILSSON, N. J. (1971). *Problem Solving in Artificial Intelligence*,

DE CHAMPEAUX, D., and SINT, L. (1975). *An Improved Bi-Directional Heuristic Search Algorithm*, Journal A.C.M., in press.

Book reviews

An Introduction to Data Structures with Applications, by J.-P. Tremblay and P. G. Sorenson, 1976; 704 pages. (*McGraw-Hill*, £12·20)

This welcome addition to the growing number of books on data structures deserves to become a classic. Acknowledging the first great contribution made by Donald Knuth (1968), the authors have given us a book more suited for use in the teaching environment, and maybe also in the industrial world of computing, since the reader is not required to delve too deeply into underlying mathematical models or their analysis. The book therefore in no way replaces Knuth's; rather it stands alongside.

The contents of the book are well matched to course I1, Data Structures of 'Curriculum 68' (1968) and to course UC1 of the 'Curriculum Recommendations for Undergraduate Programs in Information Systems' (1973). The chapter on file structures (Chapter 7—166 pages long) makes the book also relevant for courses based on UC3 (1976), as well as providing a concise introduction to the three distinct approaches currently adopted for data base systems. Chapters 3 to 6 cover linear data structures and their representation in both sequential and linked storage; nonlinear data structures; and a brief survey of the important aspects of internal sorting and searching methods. It is somewhat surprising though, to find in one book, major discussions of file structures and internal sorting methods, but little or no reference to the methods of sorting on tape or disc. The other notable chapter deals with the representation and manipulation of strings. Although the theoretical discussion of Markov Algorithms was given by Berztiss (1971) some time ago, Tremblay and Sorenson have now made readily available the practical implications, as they have influenced SNOBOL, and a multitude of text editors. Anyone embarking on the production of their first text editor could do worse than start here.

PL/I users will find, as an additional commendation, that this is the language used to illustrate algorithms. However, readers who are not familiar with PL/I should have little difficulty in following the text, since most algorithms are also described informally.

The book is extremely well written. All chapters and major sections within chapters contain helpful summaries of what is to follow, enabling the book to be used in what might be described as a 'random-access reference' mode. Exercises (without answers) are liberally sprinkled throughout the book. The publishers have done justice to the quality of the contents by the use of good layout and intelligent use of bold face type for identifiers, data values and program fragments. PL/I algorithms have, in general, been reproduced directly from computer listings.

To conclude, this book deserves to do for data structures what David Gries's book did for compiler writing, but with a style of publication that both deserve.

References

KNUTH, D. E. (1968). *The Art of Computer Programming, Fundamental Algorithms*, Addison-Wesley, Reading, Mass.

CACM, Vol. 11, No. 9, Pages 174-176, 1968.

CACM, Vol. 16, No. 12, Pages 727-749, 1973.

BERZTISS, A. T. (1971). *Data Structures, Theory and Practice*, Academic Press Inc, NY.

Physics of Computer Memory Devices, by S. Middlehoek, P. K. George and P. Dekker, 1976; 402 pages. (*Academic Press*, £12·50)

For many years digital computer stores used magnetic devices almost exclusively; this monopoly has recently been challenged by the introduction of cheap semiconductor stores, high density optical stores and other developments. This book gives a survey of all significant types of computer storage including delay line devices and materials, matrix stores using a variety of elements, beam accessed stores and moving surface stores.

Recent developments such as magnetic bubble stores, surface acoustic wave devices and charge transfer devices are described in addition to other well established storage techniques. The book covers rather more than the title suggests since it gives not only the physical principles of the various devices but includes their organisation and circuit details where appropriate.

The book should thus interest all computer engineers who wish to keep abreast of recent developments in digital computer storage systems and also physicists concerned with their underlying principles.

J. C. CLULEY (Birmingham)