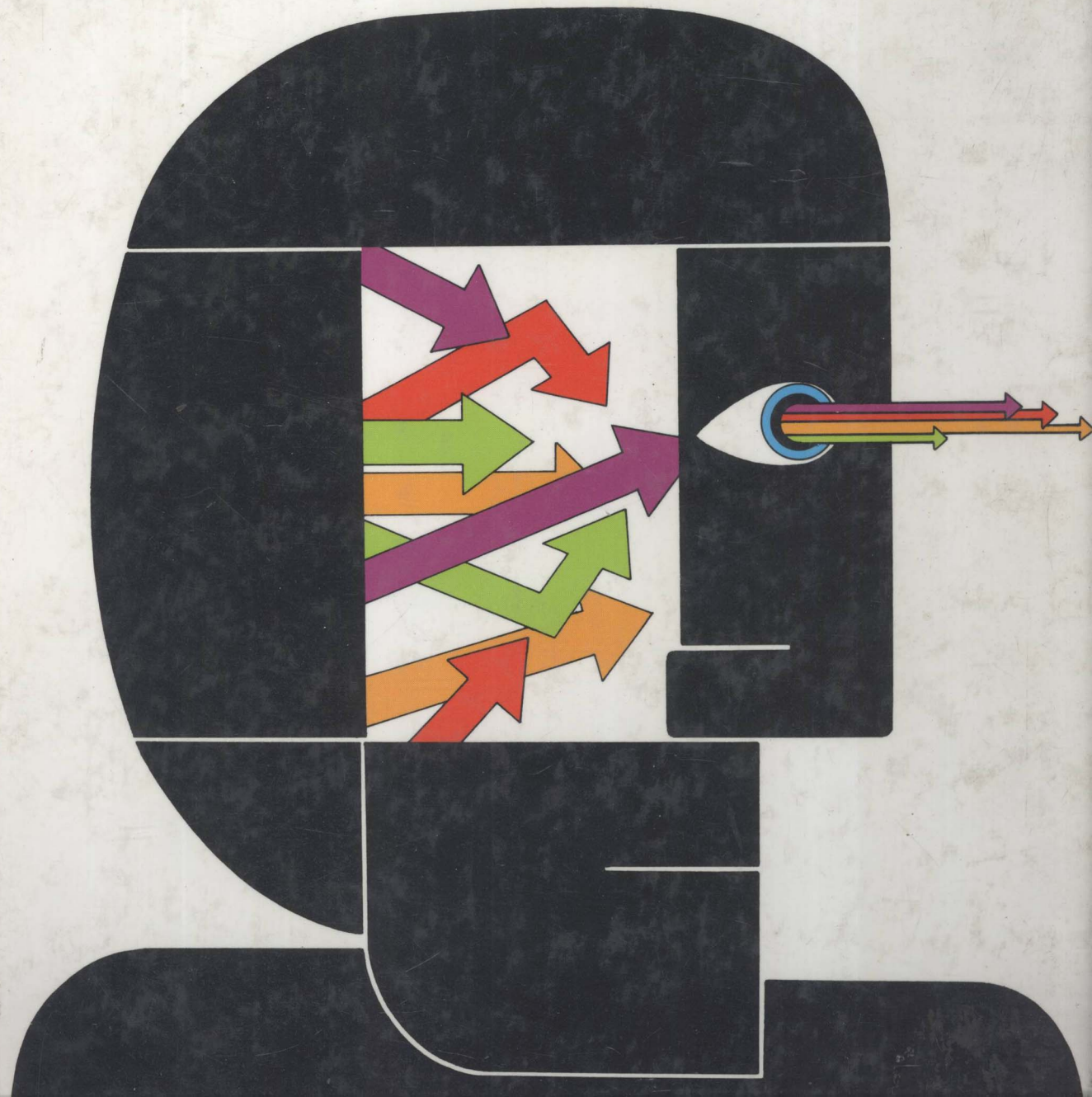
DATA STRUCTURES

FROM ARRAYS TO PRIORITY QUEUES

WAYNE AMSBURY



Data Structures

From Arrays to Priority Queues

Wayne Amsbury
Northwest Missouri State University

Wadsworth Publishing Company
Belmont, California
A Division of Wadsworth, Inc.

Computer Science Editor: *Frank Ruggirello*
Production: *Mary Forkner, Publication Alternatives*
Text Designer: *Michael Rogondino*
Copy Editor: *Joan Pendleton*
Technical Illustrator: *Carol Johnston*
Cover Design: *Stephen Osborn*
Signing Representative: *Myron Flemming*

Printed in the United States of America

1 2 3 4 5 6 7 8 9 10—89 88 87 86 85

ISBN 0-534-04590-1

Library of Congress Cataloging in Publication Data

Amsbury, Wayne, 1935–
Data structures.

Bibliography: p.
Includes index.
1. Data structures (Computer science) I. Title.
QA76.9.D35A47 1985 001.64'4 85-3159
ISBN 0-534-04590-1

Data Structures

From Arrays to Priority Queues

Wadsworth Series in Computer Information Systems

Senn, Information Systems in Management, Second Edition

Clarke and Prins, Contemporary Systems Analysis and Design

Vasta, Understanding Data Base Management Systems

Ageloff and Mojena, Applied Structured BASIC

Rob, Big Blue BASIC: Programming the IBM PC and Compatibles

Athappilly, Programming and Problem Solving in VAX-11 BASIC

Rob, Introduction to Microcomputer Programming

Amsbury, Data Structures: From Arrays to Priority Queues

Brown, From Pascal to C: An Introduction to the C Programming Language

Preface

Many application programs and discussions of topics in computer science are awkward without a relatively extensive acquaintance with data structures. Hence, the data structures course is moving rapidly downward in the curriculum. In the last few years it has changed from being a graduate-level course in analytical techniques to a foundation course for undergraduates and nonmajors. If it isn't already, data structures soon will be taught at four-year colleges as a prerequisite for most upper-level computer science courses and as a terminal computer science course in two-year programs and for many nonmajors. Consequently, the course is taught on a wide range of levels. The broadening of the audience requires some adjustment in the way a data structures course is taught.

Although its wide range of content and academic rigor make it suitable for use in higher-level courses, *Data Structures: From Arrays to Priority Queues* is intended to serve a broader audience. The book can function as an introduction to data structures quite early in the development of programming mastery. It is aimed at the student who has had one or more semesters of programming in a structured language and no specific college mathematics. At lower levels, a data structures course needs to develop algorithmic sophistication rather than exercise mathematical skills. Thus, this book keeps the comparison of data structures and algorithms in constant view, but it does not require a great deal of mathematical sophistication. *Data Structures* brings the reader to an awareness of the timing of algorithms, particularly at the level of the order of growth, but extensive analysis of algorithms applied to data structures is left to later courses. The intent is to foster an appreciation of the effectiveness of data structures and their associated algorithms, but in an introduction it is more important to construct a workable set of tools than to hone a smaller set. Data structures *themselves* are the primary focus, not their detailed analysis.

The topic of data structures plays much the same essential role in computer science that calculus plays in mathematics. The *apparent agenda* in a data struc-

tures course is the comparative study of alternate ways to solve and implement program solutions to problems. The *hidden agenda* is the development of an understanding of the use of abstraction in problem solving. This book focuses on the hidden agenda as a foundation for study of the apparent agenda. Skill in abstraction is developed gradually, as a natural part of problem solving with programs. Three features that explicitly address the hidden agenda are:

1. Use of Pascal-like pseudocode
2. Comparison of algorithmic and implementation variations
3. Generalized algorithms for data structure traverses

A form of pseudocode is used to present algorithms for several reasons. One reason is simply that no one language is ideal for all purposes, is used universally, or is dominant forever. A number of languages, including Algol, Pascal, ADA, and Modula-2, have common features extracted here. Another reason for presenting algorithms in pseudocode for the reader to translate into a working language is that the task requires some effort, but not too much. Translation requires a choice of parameter-passing classes for procedure variables and attention to some other details. Translating pseudocode does not, however, require the time and effort needed to *derive* an algorithm. An extensive amount of conceptual material can be presented succinctly in pseudocode. Use of pseudocode thus provides an appropriate balance between covering the maximum amount of ground and providing in-depth understanding of the details of the geography that is traversed. Use of pseudocode, plus examples in the text, also helps readers to develop skill in abstraction and understanding of shell-structuring and modularity.

The management of data structures involves pitfalls that lead to subtle bugs in programs, and such potential problems are best exposed by being viewed from several perspectives. Therefore, *Data Structures* emphasizes the variety of solutions to problems. For example, general traverse algorithms are developed for lists, binary trees, and priority queues. Such general algorithms provide a framework within which variations can be created and problems can be solved. The algorithms can be tailored to specific applications, and the exercise of doing so has merit in itself. The general algorithms are better guides to the variations needed for applications than is one specific variation that often fails to fit. General algorithms also promote the development of skill in abstraction in a relatively gentle way.

A separate collection of expansion and enrichment sections is included in Part II. The sections introduce or explore in depth topics that are related to the mainstream of the text. Some of these sections contain substantial working Pascal programs, initially developed in pseudocode, which provide examples of program development. The instructor thus has options for major examples, enrichment, and independent reading assignments, yet because the additional sections are separate, they do not distract from the main content.

To help the instructor tailor the course to students with varied backgrounds, most topics in the book can be covered at a variety of levels:

1. Pictorial discussion and general comprehension
2. Tracing of the crucial process(es) acting on a specific example
3. Pseudocode detailing of an algorithm with discussion of problems and variations
4. Program assignments based on class discussion
5. Program assignments based on reading
6. Experimentation with programs and analysis of the results

Potential assignments are separated into exercises, problems, programs, and projects, and the amount of effort required increases in that order. Possible assignments are mentioned in the text where they are appropriate, but they are grouped together at the end of the chapter so they can be found easily at any time. Summaries offer the reader an overview of material in each chapter.

The instructor's manual includes answers to the exercises and problems and a Pascal solution to a program assignment from each chapter.

The flexibility designed into this text has allowed it to be used at NWMSU for classes with students of mixed backgrounds, many of whom are second-semester freshmen. The central thrust of the course is coverage of the sections in Part One of this text that are not marked *optional*. In that course, Chapter 7 is considered remedial but is available for self-study, the timing-function portion of Chapter 1 is the only part of that chapter requiring class time, and Chapter 10 is seldom reached. Some optional sections in Part I and various sections of Part II are selected by instructors for supplementary readings, assignments, and sometimes class discussion.

Thorough coverage of the entire text requires a second semester or a very fast pace for even well-prepared students. The intent of the rich offering of Part II and the optional sections in Part I is to support flexibility through the selection of sections and levels of coverage. (Besides, students *will* explore accessible material on their own.)

The reviewers listed below made many suggestions that were incorporated into this book and improved it immeasurably. They are not, however, responsible for the remaining errors, whether inadvertant or stubborn: Cathy Dickerson, Black Hawk College; William H. Ford, University of the Pacific; Ken Friedenbach, University of Santa Clara; Judith L. Gersting, Indiana University-Purdue University; Henry Gordon, Kutztown State College; Nancy Griffeth, Georgia Institute of Technology; Greg Jones, Providence, Utah; Leonard Larsen, University of Wisconsin, Eau Claire; Louise Moyer, California State University, Hayward; Ron Peterson, Weber State College; Douglas Re, San Francisco Community College; Dean Sanders, Illinois State University.

Two groups of people aided and abetted this work in ways that must remain untold in detail but are appreciated in depth. One group is a team assembled by Wadsworth that includes: Myron Flemming, Frank Ruggirello, Serina Beauparlant, Mary Forkner, and Joan Pendleton. The other group consists of colleagues at

NWMSU: Merry McDonald, Gary McDonald, Robert Franks, Hong-Shi Yuan, Phil Heeler, and Margaret Adams.

My brother, David, put considerable effort into improving my technical writing in the early stages of this book, and I can only hope that the effect carried through the project.

Finally, this book could not have been produced if my wife, Carlene, had not taken over virtually all of the management of a busy household while working full time.

Wayne Amsbury

Contents

PART ONE

The Mainstream