

# Preface

This book is intended to survey the most important algorithms in use on computers today and to teach fundamental techniques to the growing number of people who are interested in becoming serious computer users. It is appropriate for use as a textbook for a second, third or fourth course in computer science: after students have acquired some programming skills and familiarity with computer systems, but before they have specialized courses in advanced areas of computer science or computer applications. Additionally, the book may be useful as a reference for those who already have some familiarity with the material, since it contains a number of computer implementations of useful algorithms.

The book consists of forty chapters which are grouped into seven major parts: mathematical algorithms, sorting, searching, string processing, geometric algorithms, graph algorithms and advanced topics. A major goal in the development of this book has been to bring together the fundamental methods from these diverse areas, in order to provide access to the best methods that we know for solving problems by computer for as many people as possible. The treatment of sorting, searching and string processing (which may not be covered in other courses) is somewhat more complete than the treatment of mathematical algorithms (which may be covered in more depth in applied mathematics or engineering courses), or geometric and graph algorithms (which may be covered in more depth in advanced computer science courses). Some of the chapters involve introductory treatment of advanced material. It is hoped that the descriptions here can provide students with some understanding of the basic properties of fundamental algorithms such as the FFT or the simplex method, while at the same time preparing them to better appreciate the methods when they learn them in advanced courses.

The orientation of the book is towards algorithms that are likely to be of practical use. The emphasis is on teaching students the tools of their trade to the point that they can confidently implement, run and debug useful algorithms. Full implementations of the methods discussed (in an actual programming language) are included in the text, along with descriptions of the operations of these programs on a consistent set of examples. Though not emphasized, connections to theoretical computer science and the analysis of algorithms are not ignored. When appropriate, analytic results are discussed to illustrate why certain algorithms are preferred. When interesting, the relationship of the practical algorithms being discussed to purely theoretical results is described. More information of the orientation and coverage of the material in the book may be found in the Introduction which follows.

One or two previous courses in computer science are recommended for students to be able to appreciate the material in this book: one course in

programming in a high-level language such as Pascal, and perhaps another course which teaches fundamental concepts of programming systems. In short, students should be conversant with a modern programming language and have a comfortable understanding of the basic features of modern computer systems. There is some mathematical material which requires knowledge of calculus, but this is isolated within a few chapters and could be skipped.

There is a great deal of flexibility in the way that the material in the book can be taught. To a large extent, the individual chapters in the book can each be read independently of the others. The material can be adapted for use for various courses by selecting perhaps thirty of the forty chapters. An elementary course on “data structures and algorithms” might omit some of the mathematical algorithms and some of the advanced graph algorithms and other advanced topics, then emphasize the ways in which various data structures are used in the implementation. An intermediate course on “design and analysis of algorithms” might omit some of the more practically-oriented sections, then emphasize the identification and study of the ways in which good algorithms achieve good asymptotic performance. A course on “software tools” might omit the mathematical and advanced algorithmic material, then emphasize means by which the implementations given here can be integrated for use into large programs or systems. Some supplementary material might be required for each of these examples to reflect their particular orientation (on elementary data structures for “data structures and algorithms,” on mathematical analysis for “design and analysis of algorithms,” and on software engineering techniques for “software tools”); in this book, the emphasis is on the algorithms themselves.

At Brown University, we’ve used preliminary versions of this book in our third course in computer science, which is prerequisite to all later courses. Typically, about one-hundred students take the course, perhaps half of whom are majors. Our experience has been that the breadth of coverage of material in this book provides an “introduction to computer science” for our majors which can later be expanded upon in later courses on analysis of algorithms, systems programming and theoretical computer science, while at the same time providing all the students with a large set of techniques that they can immediately put to good use.

The programming language used throughout the book is Pascal. The advantage of using Pascal is that it is widely available and widely known; the disadvantage is that it lacks many features needed by sophisticated algorithms. The programs are easily translatable to other modern programming languages, since relatively few Pascal constructs are used. Some of the programs can be simplified by using more advanced language features (some not available in Pascal), but this is true less often than one might think. A goal of this book is to present the algorithms in as simple and direct form as possible.

The programs are not intended to be read by themselves, but as part of the surrounding text. This style was chosen as an alternative, for example, to having `inline` comments. Consistency in style is used whenever possible, so that programs which are similar, look similar. There are 400 exercises, ten following each chapter, which generally divide into one of two types. Most of the exercises are intended to test students' understanding of material in the text, and ask students to work through an example or apply concepts described in the text. A few of the exercises at the end of each chapter involve implementing and putting together some of the algorithms, perhaps running empirical studies to learn their properties.

### *Acknowledgments*

Many people, too numerous to mention here, have provided me with helpful feedback on earlier drafts of this book. In particular, students and teaching assistants at Brown have suffered through preliminary versions of the material in this book over the past three years. Thanks are due to Trina Avery, Tom Freeman and Janet Incerpi, all of whom carefully read the last two drafts of the book. Janet provided extensive detailed comments and suggestions which helped me fix innumerable technical errors and omissions; Tom ran and checked the programs; and Trina's copy editing helped me make the text clearer and more nearly correct.

Much of what I've written in this book I've learned from the teaching and writings of Don Knuth, my thesis advisor at Stanford. Though Don had no direct influence at all on this work, his presence may be felt in the book, for it was he who put the study of algorithms on a scientific footing that makes a work such as this possible.

Special thanks are due to Janet Incerpi who initially converted the book into  $\text{\TeX}$  format, added the thousands of changes I made after the "last draft," guided the files through various systems to produce printed pages and even wrote the scan conversion routine for  $\text{\TeX}$  that we used to produce draft manuscripts, among many other things.

The text for the book was typeset at the American Mathematical Society; the drawings were done with pen-and-ink by Linda Sedgewick; and the final assembly and printing were done by Addison-Wesley under the guidance of Jim DeWolf. The help of all the people involved is gratefully acknowledged.

Finally, I am very thankful for the support of Brown University and INRIA where I did most of the work on the book, and the Institute for Defense Analyses and the Xerox Palo Alto Research Center, where I did some work on the book while visiting.

*Robert Sedgewick  
Marly-le-Roi, France  
February, 1983*