

[PDF] Algorithms In C++ Part 5: Graph Algorithms (3rd Edition) (Pt.5)

Robert Sedgewick - pdf download free book



Books Details:

Title: Algorithms in C++ Part 5: Gra
Author: Robert Sedgewick
Released: 2002-01-06
Language:
Pages: 528
ISBN: 0201361183
ISBN13: 978-0201361186
ASIN: 0201361183

[CLICK HERE FOR DOWNLOAD](#)

pdf, mobi, epub, azw, kindle

Description:

Algorithms in C is a comprehensive repository of algorithms, complete with code. If you're in a pinch and need to code something up fast, this book is the place to look. Starting with basic data structures, *Algorithms in C* covers an enormous scope of information, with extensive treatment of searching and advanced data structures, sorting, string processing, computational geometry, graph problems, and mathematical algorithms. Although the manual often neglects to provide rigorous

analysis, the text surrounding the algorithms provides clear and relevant insight into why the algorithms work. --This text refers to an out of print or unavailable edition of this title.

From the Inside Flap This book is intended to survey the most important computer algorithms in use today and to teach fundamental techniques to the growing number of people in need of knowing them. It can be used 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 taken specialized courses in advanced areas of computer science or computer applications. Additionally, the book may be useful for self-study or as a reference for those engaged in the development of computer systems or applications programs, since it contains a number of implementations of useful algorithms and detailed information on their performance characteristics. The broad perspective taken in the book makes it an appropriate introduction to the field.

SCOPE

The book contains 45 chapters grouped into 8 major parts: fundamentals, sorting, searching, string processing, geometrical algorithms, graph algorithms, mathematical algorithms and advanced topics. A major goal in developing this book has been to bring together the fundamental methods from these diverse areas, in order to provide access to the best methods known for solving problems by computer. Some of the chapters give introductory treatments of advanced material. It is hoped that the descriptions here can give readers some understanding of the basic properties of fundamental algorithms ranging from priority queues and hashing to simplex and the fast Fourier transform.

One or two previous courses in computer science or equivalent programming experience are recommended for a reader to be able to appreciate the material in this book: one course in programming in a high-level language such as C or Pascal, and perhaps another course which teaches fundamental concepts of programming systems. This book is thus intended for anyone conversant with a modern programming language and with the basic features of modern computer systems. References that might help fill in gaps in one's background are suggested in the text.

Most of the mathematical material supporting the analytic results is self-contained (or labeled as "beyond the scope" of this book), so little specific preparation in mathematics is required for the bulk of the book, though a certain amount of mathematical maturity is definitely helpful. A number of the later chapters deal with algorithms related to more advanced mathematical material--these are intended to place the algorithms in context with other methods throughout the book, not to teach the mathematical material. Thus the discussion of advanced mathematical concepts is brief, general, and descriptive.

USE IN THE CURRICULUM

There is a great deal of flexibility in how the material here can be taught. To a large extent, the individual chapters in the book can be read independently of the others, though in some cases, algorithms in one chapter make use of methods from a previous chapter. The material can be adapted for use for various courses by selecting perhaps 25 or 30 of the 45 chapters, according to the taste of the instructor and the preparation of the students.

The book begins with an introductory section on data structures and the design and analysis of algorithms. This sets the tone for the rest of the book and provides a framework within which more advanced algorithms are treated. Some readers may skip or skim this section; others may learn the basics there.

An elementary course on "data structures and algorithms" might omit some of the mathematical algorithms and some of the advanced topics, then emphasize how various data structures are used in the implementations. 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 algorithms achieve good asymptotic performance. A course on "software tools" might omit the mathematical and algorithmic material, then emphasize how to integrate the implementations given here into large programs or systems. A course on "algorithms" might take a survey approach and introduce concepts from all these areas.

Some instructors may wish to add supplementary material to the courses described above to reflect their particular orientation. For "data structures and algorithms", extra material on basic data structures could be taught; for "design and analysis of algorithms," more mathematical analysis could be added; and for "software tools," software engineering techniques could be covered in more depth. In this book, attention is paid to all these areas, but the emphasis is on the algorithms themselves.

Earlier versions of this book have been used in recent years at scores of colleges and universities around the country as a text for the second or third course in computer science and as supplemental reading for other courses. At Princeton, our experience has been that the breadth of coverage of material in this book provides our majors with an introduction to computer science that 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.

There are 450 exercises, ten following each chapter, that generally divide into two types. Most 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 them, however, involve implementing and putting together some of the algorithms, perhaps running empirical studies to compare algorithms and to learn their properties.

ALGORITHMS OF PRACTICAL USE

The orientation of the book is toward algorithms 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 are included in the text, along with descriptions of the operations of these programs on a consistent set of examples. Indeed, as discussed in the epilog, hundreds of figures are included in the book that have been created by the algorithms themselves. Many algorithms are brought to light on an intuitive level throughout the visual dimension provided by these figures.

Characteristics of the algorithms and situations in which they might be useful are discussed in detail. Though not emphasized, connections to the analysis of algorithms and theoretical computer science are not ignored. When appropriate, empirical and 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. Specific information on performance characteristics of algorithms is encapsulated throughout the "properties," important facts about the algorithms that deserve further study.

While there is little direct treatment of specific uses of the algorithms in science and engineering applications, the potential for such use is mentioned when appropriate. Our experience has been that when students learn good algorithms in a computer science context early in their education, they are able to apply them to solve problems when warranted later on.

PROGRAMMING LANGUAGE

The programming language used throughout the book is C (a Pascal version of the book is also available). Any particular language has advantages and disadvantages; we use C because it is widely available and provides the features needed for our implementations. The programs can easily be translated to other modern programming languages, since relatively few C constructs are used. Indeed, many of the programs have been translated from Pascal and other languages, though we try to use standard C idioms when appropriate.

Some of the programs can be simplified by using more advanced language features, but this is true less often than one might think. Although language features are discussed when appropriate, this book is not intended to be a reference work on C programming. When forced to make a choice, we concentrate on the algorithms, not implementation details.

A goal of this book is to present the algorithms in as simple and direct a form as possible. The programs are intended to be read not by themselves, but as part of the surrounding text. This style was chosen as an alternative, for example, to having inline comments. The style is consistent whenever possible, so that programs that are similar look similar.

ACKNOWLEDGEMENTS

Many people gave me helpful feedback on earlier versions of this book. In particular, students at Princeton and Brown suffered through preliminary versions of the material in this book in the 1980's. Special thanks are due to Trina Avery, Tom Freeman, and Janet Incerti for their help in producing the first edition. I would particularly like to thank Janet for converting the book into TeX format, adding the thousands of changes I made after the "last draft" of the first edition, guiding the files through various systems to produce printed pages and even writing the scan-conversion routine for TeX used to produce draft manuscripts, among many other things. Only after performing many of these tasks myself for later versions do I truly appreciate Janet's contribution. I would also like to thank the many readers who provided me with detailed comments about the second edition, including Guy Almes, Jay Gischer, Kennedy Lemke, Udi Manber, Dana Richards, John Reif, M. Rosenfeld, Stephen Seidman, and Michael Quinn.

Many of the designs in the figures are based on joint work with Marc Brown in the "electronic classroom" project at Brown University in 1983. Marc's support and assistance in creating the designs (not to mention the system with which we worked) are gratefully acknowledged. I also would like to acknowledge Sarantos Kapidakis' help in producing the endpapers.

This C version owes its existence to the persistent questions of several readers about C code for ALGORITHMS and to the support of Keith Wollman at Addison-Wesley, who convinced me to proceed. Dave Hanson's willingness to answer questions about ANSI C was invaluable. I also would like to thank Darcy Cotten and Skip Plank for their help in producing the book, and Steve Beck for finding the "last bug" in the printing software.

Much of what I've written here I've learned from the teaching and writings of Don Knuth, my advisor at Stanford. Though Don had no direct influence 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.

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. Many parts of the book are dependent on research that has been

generously supported by the National Science Foundation and the Office of Naval Research. Finally, I would like to thank Bill Bowen, Aaron Lemonick, and Neil Rudenstine at Princeton University for their support in building an academic environment in which I was able to prepare this book, despite numerous other responsibilities. 0201514257P04062001 --This text refers to an out of print or unavailable edition of this title.

- Title: Algorithms in C++ Part 5: Graph Algorithms (3rd Edition) (Pt.5)
 - Author: Robert Sedgewick
 - Released: 2002-01-06
 - Language:
 - Pages: 528
 - ISBN: 0201361183
 - ISBN13: 978-0201361186
 - ASIN: 0201361183
-