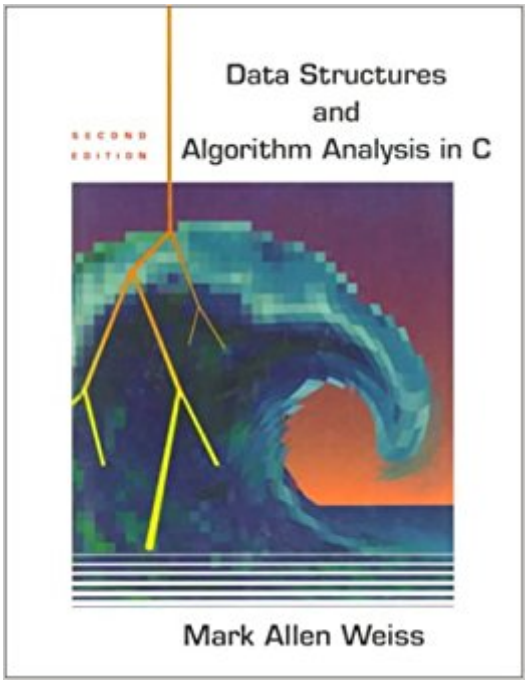
Data Structures And Algorithm Analysis In C (2nd Edition)



Synopsis

In this second edition of his best-selling book, Data Structures and Algorithm Analysis in C, Mark Allen Weiss, continues to refine and enhance his innovative approach to algorithms and data structures. Using a C implementation, he highlights conceptual topics, focusing on ADTs and the analysis of algorithms for efficiency as well as performance and running time. Dr. Weiss also distinguishes Data Structures and Algorithm Analysis in C with the extensive use of figures and examples showing the successive stages of an algorithm, his engaging writing style, and a logical organization of topics.

Book Information

Paperback: 600 pages

Publisher: Pearson; 2 edition (September 19, 1996)

Language: English

ISBN-10: 0201498405

ISBN-13: 978-0201498400

Product Dimensions: 7.5 x 1.1 x 8.9 inches

Shipping Weight: 2.4 pounds (View shipping rates and policies)

Average Customer Review: 3.2 out of 5 stars 33 customer reviews

Best Sellers Rank: #163,567 in Books (See Top 100 in Books) #30 inÃ Â Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Structured Design #30 inÃ Â Books > Computers & Technology > Programming > Algorithms > Data Structures #49 inÃ Â Books > Textbooks > Computer Science > Algorithms

Customer Reviews

Purpose/GoalsThis book describes data structures, methods of organizing large amounts of data, and algorithm analysis, the estimation of the running time of algorithms. As computers become faster and faster, the need for programs that can handle large amounts of input becomes more acute. Paradoxically, this requires more careful attention to efficiency, since inefficiencies in programs become most obvious when input sizes are large. By analyzing an algorithm before it is actually coded, students can decide if a particular solution will be feasible. For example, in this text students look at specific problems and see how careful implementations can reduce the time constraint for large amounts of data from 16 years to less than a second. Therefore, no algorithm or data structure is presented without an explanation of its running time. In some cases, minute details that affect the running time of the implementation are explored.Once a solution method is

determined, a program must still be written. As computers have become more powerful, the problems they must solve have become larger and more complex, requiring development of more intricate programs. The goal of this text is to teach students good programming and algorithm analysis skills simultaneously so that they can develop such programs with the maximum amount of efficiency.This book is suitable for either an advanced data structures (CS7) course or a first-year graduate course in algorithm analysis. Students should have some knowledge of intermediate programming, including such topics as pointers and recursion, and some background in discrete math. ApproachI believe it is important for students to learn how to program for themselves, not how to copy programs from a book. On the other hand, it is virtually impossible to discuss realistic programming issues without including sample code. For this reason, the book usually provides about one-half to three-quarters of an implementation, and the student is encouraged to supply the rest. Chapter 12, which is new to this edition, discusses additional data structures with an emphasis on implementation details.The algorithms in this book are presented in ANSI C, which, despite some flaws, is arguably the most popular systems programming language. The use of C instead of Pascal allows the use of dynamically allocated arrays (see, for instance, rehashing in Chapter 5). It also produces simplified code in several places, usually because the and (&&) operations is short-circuited.Most criticisms of C center on the fact that it is easy to write code that is barely readable. Some of the more standard tricks, such as the simultaneous assignment and testing against 0 via if (x=y) are generally not used in the text, since the loss of clarity is compensated by only a few keystrokes and no increased speed. I believe that this books demonstrates that unreadable code can be avoided by exercising reasonable care.OverviewChapter 1 contains review material on discrete math and recursion. I believe the only way to be comfortable with recursion is to see good uses over and over. Therefore, recursion is prevalent in this text, with examples in every chapter except Chapter 5.Chapter 2 deals with algorithm analysis. This chapter explains asymptotic analysis and its major weaknesses. Many examples are provided, including an in-depth explanation of logarithms running time. Simple recursive programs are analyzed by intuitively converting them into iterative programs. More complicated divide-and-conquer programs are introduced, but some of the analysis (solving recurrence relations) is implicitly delayed until Chapter 7, where it is performed in detail.Chapter 3 covers lists, stacks, and queues. The emphasis here is on coding these data structures using ADTs, fast implementation of these data structures, and an exposition of some of their uses. There are almost no programs (just routines), but the exercises contain plenty of ideas for programming assignments.Chapter 4 covers trees, with an emphasis on search trees, including external search trees (B-trees). The UNIX file system and expression trees are used as examples.

AVL trees and splay trees are introduced but not analyzed. Seventy-five percent of the code is written, leaving similar cases to be completed by the student. More careful treatment of search tree implementation details is found in Chapter 12. Additional coverage of trees, such as file compression and game trees, is deferred until Chapter 10. Data structures for an external medium are considered as the final topic in several chapters.Chapter 5 is relatively short chapter concerning hash tables. Some analysis is performed, and extendible hashing is covered at the end of the chapter.Chapter 6 is about priority queues. Binary heaps are covered, and there is additional material on some of the theoretically interesting implementations of priority queues. The Fibonacci heap is discussed in Chapter 11, and the pairing heap is discussed in Chapter 12.Chapter 7 covers sorting. It is very specific with respect to coding details and analysis. All the important general-purpose sorting algorithms are covered and compared. Four algorithms are analyzed in detail: insertion sort, Shellsort, heapsort, and quicksort. The analysis of the average-case running time of heapsort is new to this edition. External sorting is covered at the end of the chapter.Chapter 8 discusses the disjoint set algorithm with proof of the running time. This is a short and specific chapter that can be skipped if Kruskal's algorithm is not discussed.Chapter 9 covers graph algorithms. Algorithms on graphs are interesting, not only because they frequently occur in practice but also because their running time is so heavily dependent on the proper use of data structures. Virtually all of the standard algorithms are presented along with appropriate data structures, pseudocode, and analysis of running time. To place these problems in a proper context, a short discussion on complexity theory (including NP-completeness and undecidability) is provided.Chapter 10 covers algorithm design by examining common problem-solving techniques. This chapter is heavily fortified with examples. Pseudocode is used in these later chapters so that the student's appreciation of an example algorithm is not obscured by implementation details.Chapter 11 deals with amortized analysis. Three data structures from Chapters 4 and 6 and the Fibonacci heap, introduced in this chapter, are analyzed.Chapter 12 is new to this edition. It covers search tree algorithms, the k-d tree, and the pairing heap. This chapter departs from the rest of the text by providing complete and careful implementations for the search trees and pairing heap. The material is structured so that the instructor can integrate sections into discussions from other chapters. For example, the top-down red black tree in Chapter 12 can be discussed under AVL trees (in Chapter 4).Chapters 1-9 provide enough material for most one-semester data structures courses. If time permits, then Chapter 10 can be covered. A graduate course on algorithm analysis could cover Chapters 7-11. The advanced data structures analyzed in Chapter 11 can easily be referred to in the earlier chapters. The discussion of NP-completeness in Chapter 9 is far too brief to

be used in such a course. Garey and Johnson’s book on NP-completeness can be used to augment this text.ExercisesExercises, provided at the end of each chapter, match the order in which material is presented. The last exercises may address the chapter as a whole rather than a specific section. Difficult exercises are marked with an asterisk, and more challenging exercises have two asterisks.A solutions manual containing solutions to almost all the exercises is available to instructors from the Addison-Wesley Publishing Company.References References are placed at the end of each chapter. Generally the references either are historical, representing the original source of the material, or they represent extensions and improvements to the results given in the text. Some references represent solutions to exercises.Code AvailabilityThe example program code in this book is available via anonymous ftp at aw. It is also accessible through the World Wide Web; the URL is aw/cseng/authors/weiss/dsaac2/dsaac2e.sup.html (follow the links from there). The exact location of this material may change.AcknowledgmentsMany, many people have helped me in the preparation of books in this series. Some are listed in other versions of the book; thanks to all.For this edition, I would like to thank my editors at Addison-Wesley, Carter Shanklin and Susan Hartman. Teri Hyde did another wonderful job with the production, and Matthew Harris and his staff at Publication Services did their usual fine work putting the final pieces together.M.A.W. Miami, Florida July, 19960201498405P04062001

In this second edition of his best-selling book, Data Structures and Algorithm Analysis in C, Mark Allen Weiss, continues to refine and enhance his innovative approach to algorithms and data structures. Using a C implementation, he highlights conceptual topics, focusing on ADTs and the analysis of algorithms for efficiency as well as performance and running time. Dr. Weiss also distinguishes Data Structures and Algorithm Analysis in C with the extensive use of figures and examples showing the successive stages of an algorithm, his engaging writing style, and a logical organization of topics. Features Includes a chapter on algorithm and design techniques that covers greedy algorithms, divide and conquer algorithms, dynamic programming, randomized algorithms, and backtracking Presents current topics and newer data structures such as Fibonacci heaps, skew heaps, binomial queues, skip lists, and splay trees Contains a chapter on amortized analysis that examines the advanced data structures presented earlier in the book Provides a new chapter on advanced data structures and their implementation covering red black trees, top down splay trees, treaps, k-d trees, pairing heaps, and more Incorporates new results on the average case analysis of heapsort Offers source code from example programs via anonymous FTP 0201498405B04062001

I borrowed this book from my workplace library and spent a couple of months working through it - This text is first rate, the discussion is lively and engaging and the problems are so good for building up your algorithm design and problem solving abilities. I had to order a copy for myself - to keep as a reference.Data structures and Algorithm analysis is not an easy subject to master, you have to be prepared for work hard, and do lots of problems before you really appreciate the subject. There are lots of books out there, which try to dumb down the subject - I believe that is a false economy of effort. This author does not take that approach, he instead presents a treatment of Data structures and algorithm which sets you on the course to mastery.I can’t stress enough how much systematically working through the problems in this book, helped my in my day to day programming life.

Works, but I prefer a book with answers to the exercises especially for beginners.

I recently checked this book out at my college’s library,when I realized that I absolutely must have a copy of myown. The book contains all the information that I wouldneed for the annual ACM programming contest, and as an addedplus, its written with regards to my programming languageof choice, C.A must have reference.

The condition of the book was as expected and received it within the expected time. No problems transaction with smooth. Would purchase from the seller again.

This is the textbook for my undergraduate data structures class. In my opinion this book is more suited to graduate level study, or for very experienced C programmers. The author assumes that you have a very solid grounding in C language structure and discrete math, his examples are often cryptic and incomplete leaving it up to the reader to figure out what is meant. In my opinion this book is not well suited for learning the fundamentals of data structures, and I’m still looking for one that is.

Many people on here have complained that this book goes way over the head of students not already familiar with some C++ and data structures. To these comments, I refer you to the product description (or editorial review, whichever) that specifically says this is an advanced text. I apologize to those whose professors ordered this book for intro data structures--I can understand why this book would go past the scope of that class. However, if you know any Object Oriented programming

(Java or C++ preferably) and know some basic algorithms and structures (matrices, sorts, recursion, trees, queues, etc.) this book will take you far. I don't even mean that you must be proficient in these structures, just have some basic understanding of how they work. For example, you should know what a tree is (root, leaves, implemented with pointers and nodes) and book will tell you how to use trees (B-Trees, Binary trees, etc.). By the point you are using this book, hopefully you'll have taken the math and programming classes needed to comprehend this text. Otherwise, do not blame the text for being targeted to an advanced audience.

This book was used in CS 303 at the University of Alabama at Birmingham.It is overall a good book and covers many topics in a reasonable depth. Its very readable and definitely worth the buy. The *best* part are the supplied references at the end of each chapter.My complaints:1. Focuses on a Specific language. While there are editions for a number of languages, Weiss does not give pseudo code implementations, so you have to rely on your programming knowledge. While this is beneficial since it grounds the details in a real language, I'd rather not get in the details at this point. And things like necessary operations for data structures are provided only in code form, so you have to read and understand the code. That is more a practical concern that shouldnt be focused on. I prefer a higher level, more mathematically intense analysis.2. No answer to questions asked at the end of the section. This would be nice, especially for independent study, but not overly needed in this book. Its written well enough that it is not as much of a requirement.3. The mathematical treatment of the algorithm analysis is a bit lacking. What little there is is not well explained. For those without the needed background, its useless, for those with a moderate background a sufficient amount of detail is not provided.

As a computer science student having this book for (dinner) my course in structures and algorithms, my comment will not be of the fool proof theoretical academical type.I find this book very useful.It has a lot of code examples, and in my oppinion it is perfectfor those who has some experience writing C++ code.The implementations rely heavily on templates, which(will effectively scare away the remaining students)is actualy irrelevant when it comes to most of the algorithms.I say this even if our course only covered 60-70 % of the book.Luckily for me, I already loved templates when I started the course, but I dont think this was the case for most of my fellow students.The book is vell organized, and it has a lot of "easy to understand" drawings all the way through.It starts with a tutorial on advanced C++ topics for those who just finished their ABC book in C++ programming(like Deitel&Deitel).The code examples are very professional, tight and bug free.If you are happy writing

C code, this book is not for you.There is some use of STL througout the book, but it does not require you to be an expert on the topic.I think it has a deep and thorough examination of all the topics, and it covers more structures and algorithms I could dream of for at least the next 2 years.I recommend this book to all C++ code writing engineer students(who are not afraid of irrelevant templates)

Analytics: Data Science, Data Analysis and Predictive Analytics for Business (Algorithms, Business Intelligence, Statistical Analysis, Decision Analysis, Business Analytics, Data Mining, Big Data) Data Structures and Algorithm Analysis in C (2nd Edition) Data Structures and Algorithm Analysis in Java (2nd Edition) Data Analytics: What Every Business Must Know About Big Data And Data Science (Data Analytics for Business, Predictive Analysis, Big Data Book 1) Data Analytics: Applicable Data Analysis to Advance Any Business Using the Power of Data Driven Analytics (Big Data Analytics, Data Science, Business Intelligence Book 6) Data Structures and Algorithm Analysis in Java (3rd Edition) Data Structures and Algorithm Analysis in C++ (3rd Edition) Data Structures & Algorithm Analysis in C++ Analytics: Business Intelligence, Algorithms and Statistical Analysis (Predictive Analytics, Data Visualization, Data Analytics, Business Analytics, Decision Analysis, Big Data, Statistical Analysis) Big Data For Business: Your Comprehensive Guide to Understand Data Science, Data Analytics and Data Mining to Boost More Growth and Improve Business - Data Analytics Book, Series 2 Data Analytics For Beginners: Your Ultimate Guide To Learn and Master Data Analysis. Get Your Business Intelligence Right Ã¢â ¬â œ Accelerate Growth and Close More Sales (Data Analytics Book Series) Java Software Structures: Designing and Using Data Structures (4th Edition) Starting Out with Java: From Control Structures through Data Structures (3rd Edition) Data Structures and Algorithms Made Easy in Java: Data Structure and Algorithmic Puzzles Algorithm Design: Foundations, Analysis, and Internet Examples Design and Analysis of Composite Structures: With Applications to Aerospace Structures Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data Data Analytics and Python Programming: 2 Bundle Manuscript: Beginners Guide to Learn Data Analytics, Predictive Analytics and Data Science with Python Programming Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking Discovering Knowledge in Data: An Introduction to Data Mining (Wiley Series on Methods and Applications in Data Mining)

Privacy

FAQ & Help