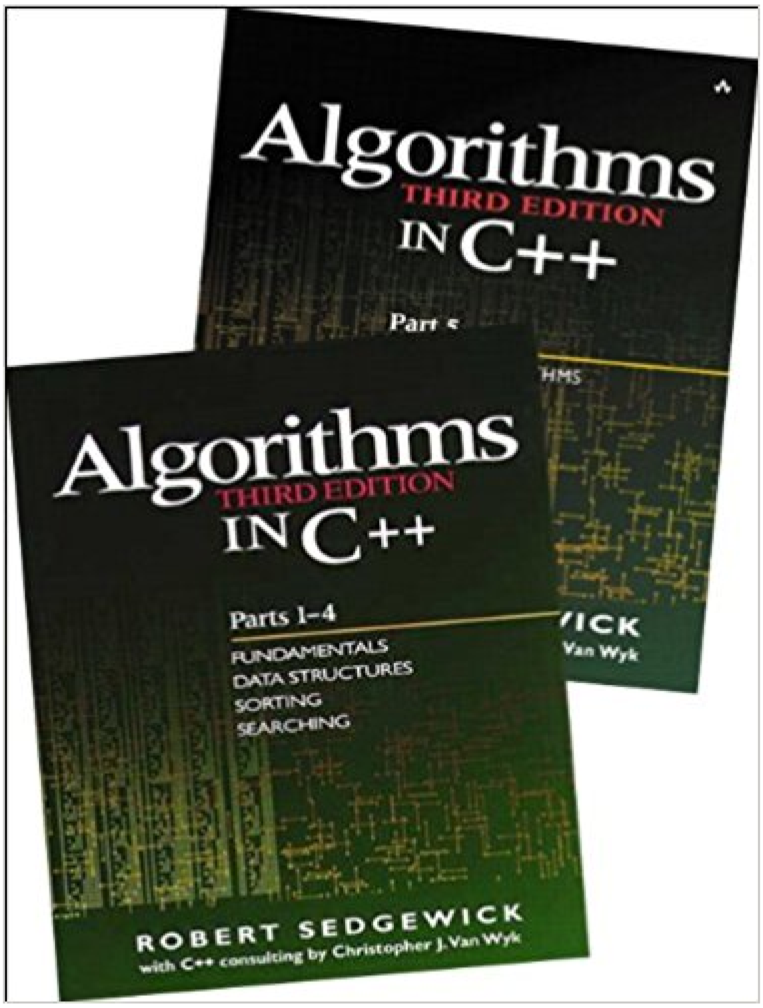
Bundle Of Algorithms In C++, Parts 1-5: Fundamentals, Data Structures, Sorting, Searching, And Graph Algorithms (3rd Edition) (Pts. 1-5)



Synopsis

Software developers and computer scientists have eagerly awaited this comprehensive revision of Robert Sedgewick’s landmark texts on algorithms for C++. Sedgewick has completely revamped all five sections, illuminating today’s best algorithms for an exceptionally wide range of tasks. This shrink-wrapped package brings together Algorithms in C++, Third Edition, Parts 1-4 and his new Algorithms in C++, Third Edition, Part 5. Together, these books are the most definitive, up-to-date, and practical algorithms resource available. The first book introduces fundamental concepts associated with algorithms; then covers data structures, sorting, and searching. The second book focuses entirely on graphing algorithms, which are critical for a wide range of applications, including network connectivity, circuit design, scheduling, transaction processing, and resource allocation. Sedgewick focuses on practical applications, giving readers all the information, diagrams, and real (not pseudo-) code they need to confidently implement, debug, and use the algorithms he presents. Together, these books present nearly 2,000 new exercises, hundreds of new figures, and dozens of new programs.

Book Information

Paperback: 1200 pages

Publisher: Addison-Wesley Professional; 3 edition (January 7, 2002)

Language: English

ISBN-10: 020172684X

ISBN-13: 978-0201726848

Product Dimensions: 7.9 x 2.4 x 9.1 inches

Shipping Weight: 4.5 pounds (View shipping rates and policies)

Average Customer Review: 3.2 out of 5 stars 7 customer reviews

Best Sellers Rank: #352,192 in Books (See Top 100 in Books) #53 inÃ Â Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Structured Design #58 inÃ Â Books > Computers & Technology > Programming > Algorithms > Data Structures #96 inÃ Â Books > Textbooks > Computer Science > Algorithms

Customer Reviews

Software developers and computer scientists have eagerly awaited this comprehensive revision of Robert Sedgewick’s landmark texts on algorithms for C++. Sedgewick has completely revamped all five sections, illuminating today’s best algorithms for an exceptionally wide range of tasks. This shrink-wrapped package brings together Algorithms in C++, Third Edition, Parts 1-4 and his new

Algorithms in C++, Third Edition, Part 5, at a special discounted price. Together, these books are the most definitive, up-to-date, and practical algorithms resource available. The first book introduces fundamental concepts associated with algorithms, then covers data structures, sorting, and searching. The second book focuses entirely on graphing algorithms, which are critical for a wide range of applications, including network connectivity, circuit design, scheduling, transaction processing, and resource allocation. Sedgewick focuses on practical applications, giving readers all the information, diagrams, and real (not pseudo-) code they need to confidently implement, debug, and use the algorithms he presents. Together these books present nearly 2,000 new exercises, hundreds of new figures, and dozens of new programs.   020172684XB07112001

Robert Sedgewick is the William O. Baker Professor of Computer Science at Princeton University. He is a Director of Adobe Systems and has served on the research staffs at Xerox PARC, IDA, and INRIA. He earned his Ph.D from Stanford University under Donald E. Knuth.
020172684XAB06262002

First of all, let me tell you on advance that I have not finished the book, but I am not nearly close to the "first pages" either: I think I have read enough of it to give it a good evaluation. I will post updates later, as need arises. UPDATE: I have already finished the book.I don’t know. Maybe it’s because its presentation, the way the author narrates, or just because the renown of the book makes you biased, but as soon as you start reading the very first pages you know you are in for something very special. You know that, after reading this book, you will be a much better computer scientist. I will not lie: I am still in attending college, but I have always been a great programmer, and in all my programming classes I am top of the class. I was really full of myself before buying this book. How wrong was I!The first thing you get after the preface is some advices on the exercises. The author tells you how some of them are really hard, or some are just really easy, etc. He also states that "There are far too many exercises for you to do read and assimilate them all..." When I read this, I gleaned each section and the amount of exercises (there are literally, thousands), and I said to myself: "I will work on, at the least, 80% of the exercises on each section. I will never avoid the hard or very hard ones" (Sedgewick explicitly tells you how to differentiate them). Alas, even some exercises that are not considered to be hard have been hell on me. Until know, I have been true to my word, but I will get into that later.Of course, this book is not about the exercises, is about Algorithms, and I’d be dammed if Sedgewick is not considered among the best authors of all times on the topic. I’ve read other books on Algorithms, and I always find that most books fit into one of these two spectrums: either they are recipes, or they are strictly academic, with little practical use. Algorithms in C++ is neither. Sedgewick makes a deep and interesting analysis on the properties of each algorithm (and data structure), concentrating at first on its abstract properties, not even daring talk about implementations. He is extremely clear in his language. Look, I am an Ecuadorian, living in Ecuador, and my English is pretty much what I have learned in my school or by myself. I can understand Sedgewick perfectly. I don’t think I have had to reread a thing twice because the writing was obscure.After being all abstract and wonderfull and stuff, Sedgewick then implements, and he does so magnificently. Yes, he may have not the best of C++ practices (for example, he explains what C++ operator delete does, but seems to dislike the use of it), or even good programming practices at all (you have, in one program, variables named like: p, a, x, y, z, f), but his implementations are nevertheless clear, straight to the point and efficient. He also constantly reminds the reader how important is to separate implementations from abstract concepts, but also explains the advantages certain implementations have against others.So, you have learned the concept of the algorithm and it’s properties, implemented it, and just when you think it as all over, Sedgwick goes ahead and refines the abstract model of the algorithm, implements it again, and thus a refinement iteration process emerges! All this is aided by graphics, more programs, and clear and concise definitions (which are separated from the text in italics). This whole process is done in one section or many. For example, he explains Pushdown-Stacks in several sections, but using this same process. Don’t get me wrong now, Sedgwick does not use a "template" for explaining each topic: he will change his ways of explaining things if necessary, and as a whole the book reads very naturally, it never gets old or repetitive.Once you finish each section, you get a bunch of exercises. Trust me: WORK ON THEM. Especially the ones marked with unfilled circles. They will bring new insights on the topic (whereas after such grand explanations, you would think more insight could not be obtained). As I said before, I have worked, until now, on more than the 80% of all the exercises. Some require background programming experience, some a high level of math (but not never too high), and they will slow your reading down (there are loads and loads of them!), but if you do them, you will be cool. Your projects will be most efficient and elegant. Teachers will admire your work (though secretly you know they should be admiring Robert Sedgewick).One final note: this is not an academic book on the analysis of algorithms. This book is strictly for people who really want to put the best algorithms and data structures into practice. Sedgewick will tell you about the O notation, and what it means for an algorithm to run proportional to some function, but it is not the goal of the book to make you sit down and mathematically extract the efficiency of an algorithm. For this Sedgwick has another book, which people hail and glorify too. But with Algorithms in C++ (and C, and Java) you will be able to sit in front of a computer, analyze some problem (or a previously implemented solution to that problem) and determine the most effective way of solving it (or if you can replace the code with a more efficient one).After walls of text, I hide again. I really hope this helped.

These books were recommended by a professor of a data structures class I took as the kind of books that were worth keeping.The author was very thorough when he wrote this book, even to the point that he'd has a classification scheme for each question.In addition to that, I don't think I've found a typo in this book... ever. With that said, I've read the first book thoroughly but haven't read much of the second as graphing algorithms are not as important to me.I'd recommend this book to anyone with an interest in algorithms and mathematical maturity.

Great refresher and reference. Used this book in college, and lost it. Worth buying again. Great examples and exercises.

Any professional programmer would benefit from having these books at hand. Excellent discussions of the basic algorithms which every programmer needs to know.But I would like to particularly highlight the discussions on binary and n-ary search trees. The most enlightening discussion in print, giving the reader a real synoptic view of search tree algorithms, how they evolved, and their culmination in red-black trees.Other reviewers have mentioned that the algorithms as presented here seem to be just warmed=over versions of their C counterparts presented in the C edition of this work. There is a germ of truth to this, but I really don't consider it to be a valid criticism of the books. The point here is not to present C++ coding techniques, but to understand algorithms. If you want to know what a state-of-the art C++ implementation of Red-Black trees looks like, just read the source code which comes with the GNU compiler toolchain. But you're not going to have a prayer of understanding it until you first understand how Red-black trees work--that's where this book comes in. If you are trying to explain the Red-black tree algorithm, you don't want all of the C++ do-dads and optimizations, templates, etc, all cluttering up the presentation of the skeletal algorithm.

Cover has already fallen of and pages are starting to come loose. I believe this book is printed when ordered and not given sufficient time to dry

If you need a book to introduce yourself in data structures, thats not your book. This books are for consult, not to learn, cos there are leaks : insuficient code, insuficient large explanations and drawings about TDAs. Furthermore, its expensive.Even trough that, Id recommended part 5 because its a good collection of the most used algorithms based in graphs.

The examples shown are quite complicated for me and they intend to related to each other. So assume you want to understand some topic. You can not just jump into that and start understanding. In order to understand it, you may need to understand examples from previous chapters to better understand current one. The author draw several tree structures to explain the running process which only make it much difficult to understand. I totally do not get this book. Maybe it is why the can not let us read some contents inside. Once you read one chapter, you may not want to buy it.

Download to continue reading...

Designing and Using Data Structures (4th Edition) Data Structures And Algorithms Using Java Data Structures, Algorithms, and Software Principles in C Data Structures and Algorithms in Java Problem Solving with Algorithms and Data Structures Using Python

Contact Us

DMCA

Privacy

FAQ & Help