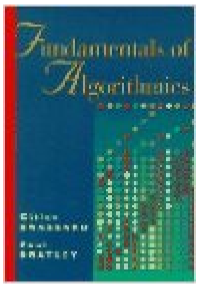
# Fundamentals Of Algorithmics



Fundamentals of Algorithmics

## Synopsis

This is an introductory-level algorithm book. It includes worked-out examples and detailed proofs. Presents Algorithms by type rather than application. Includes structured material by techniques employed, not by the application area, so readers can progress from the underlying abstract concepts to the concrete application essentials. It begins with a compact, but complete introduction to some necessary math. And it approaches the analysis and design of algorithms by type rather than by application.

## Book Information

Hardcover: 524 pages

Publisher: Pearson; 1 edition (August 20, 1995)

Language: English

ISBN-10: 0133350681

ISBN-13: 978-0133350685

Product Dimensions: 6.9 x 1.2 x 9.2 inches

Shipping Weight: 2 pounds (View shipping rates and policies)

Average Customer Review: 3.0 out of 5 starsÂ Â See all reviewsÂ (18 customer reviews)

Best Sellers Rank: #529,213 in Books (See Top 100 in Books) #123 inÂ Books > Textbooks > Computer Science > Algorithms #199 inÂ Books > Science & Math > Mathematics > Pure Mathematics > Discrete Mathematics #312 inÂ Books > Computers & Technology > Programming > Algorithms

## Customer Reviews

This book is written in a simple style: algoritm class description, related proofs, examples, variations and exemples. Because of this clean approach, you can basically read it in a couple days if you have the necessary background knowledge, and in a couple weeks if not. I strongly recommend this book to everyone who are calling themselves "programers", either graduated or not, it will teach/remind them what are the elegant fundamental algorithmic approches to common problems; it’s seems that today’s software could use a bit of this elegance. As for all the negative reviews, I’m astonished. I own several books on the subject, and I find this one the easiest to read, no question. It must be some sort of personal vendetta against the author, a failed undegraduate course with this book as teaching material, or some other personal reason; that’s probably why this/these person(s) whish(es) to remain anonymous.

One of the few books organising algorithms around design techniques instead of application areas (see "Algorithms" from Sedgevick). Though both approachs have pros and cons, I've found the former most convenient for people learning to design algorithms. Many problems admit several solutions depending on the design tecnique involved. This fact is emphasized by the authors.The book is self-contained, plainly written, so that an undergraduate can read it after taking a prerequisite course on programming.It's also mathematically rigorous, while mantaining simplicity. This allows analysis of algorithms to be really "analytic" (no more guessing the solution to a recurrence relation).One main drawback (certainly usual) is the lack of excercise solutions.

I don't agree with the other reviews. This book is a serious one and it is a "smooth" version of "Algorithms: Theory and Practice" from the same authors, which has a 5 stars rate. The book don't teach how to program, but the main kind of techniques in algorithm design (Greedy, Divide and Conquer and Dynamic Programing techniques, and others like Backtracking, Probabilistic Algorithms, etc.). I teach a course of Algorithm Design with this book (having the Sedgewick and Cormen books like additional very good references) and I think that it is a very instructive course on algorithmics (but NOT in programming, of course).

I got this book for an Algorithm class at my school, and found it to be a bit tough to read, it was very complex, and from the beginning started off with advance notation and highly mathematic language. The book does not provide nearly enough examples, to help with assignments (or problems from its review), and forced me to have to turn to the internet for easier to follow guidance. The book offers examples and plenty of proofs, which make this book nice for reference for proving a fact or referencing a paper, however, it is not very practical for use in the classroom, since it does not offer enough examples to follow to learn how to complete assignments.In the end this book offers great proofs and highly involved algorithms in pseudocode and mathematical notation, which makes this book great for reference and writing papers, however, it lacks in helping in the classroom for assignments and other items where an easy to follow example would helpful. However, for any one interested in Computer Science, I would recommend this book, for reference for algorithms, and their worst/best case analysis.

This is a good book - far better than an average rating of 2 stars would suggest. However, i think a big reason for such a rating is that it's supposed to be an introductory textbook (as claimed by the authors in the preface), and from that point of view, it leaves much to be desired. The presentation is such that it can be difficult for first-timers to follow, and many topics are simply absent, e.g. network flows. On the other hand, those already familiar with the material will find it quite readable and handy as a reference or for brushing up. The authors choose to present algorithms from a design point of view, ie greedy approach, divide-and-conquer, bactracking, dynamic programming etc instead of the traditional way of presenting algorithms for solving a class of problems, e.g searching, sorting, graph algorithms and so on. While this is good for those who're already familiar with the traditional approach and also offers a unified view of problem-solving strategies in Algorithmics, it is decidedly easier for the beginning student to have (e.g) all sorting algorithms in one place than to discover that they're scattered across different chapters. All algorithms are presented in pseudocode, and are thus have no programming language dependencies. A healthy number of solved examples have been provided, and unsolved problems are numerous. In summary, i think this is quite a good book for practitioners, researchers and those doing an advanced course in Algorithms, whereas it may not be suitable for an introductory course (except, perhaps, as a supplement).

Fundamentals of Algorithmics Fundamentals of Nursing: Human Health and Function (Craven, Fundamentals of Nursing: Human Health and Functionraven, Fundamentals of Nurs) Fundamentals of Office 365: 2016 Edition (Computer Fundamentals) Fundamentals of Hydrology (Routledge Fundamentals of Physical Geography) Tietz Fundamentals of Clinical Chemistry and Molecular Diagnostics, 7e (Fundamentals of Clinical Chemistry (Tietz)) Fundamentals of Biostatistics (Rosner, Fundamentals of Biostatics) Kozier & Erb's Fundamentals of Nursing (10th Edition) (Fundamentals of Nursing (Kozier)) Fundamentals of Geomorphology (Routledge Fundamentals of Physical Geography) Bowling Fundamentals (Sports Fundamentals) Bowling - Step By Step Guide For A Beginner To Learn The Fundamentals Of Bowling (Bowling fundamentals, Bowling Tips, Bowling Basics, Bowling Professional, Bowling Technique) Tennis Fundamentals (Sports Fundamentals) Volleyball Fundamentals (Sports Fundamentals) Racquetball Fundamentals (Sports Fundamentals) TIBCO Architecture Fundamentals (TIBCO Press) Bundle: Illustrated Microsoft Office 365 & Office 2016: Fundamentals, Loose-leaf Version + MindTap Computing, 1 term (6 months) Printed Access Card ... Office 365 & Office 2016: Introductor Database Processing: Fundamentals, Design, and Implementation (13th Edition) Database Processing: Fundamentals, Design, and Implementation, Ninth Edition Big Data, MapReduce, Hadoop, and Spark with Python: Master Big Data Analytics and Data Wrangling with MapReduce Fundamentals using Hadoop, Spark, and Python Big Data Fundamentals: Concepts, Drivers & Techniques (The Prentice Hall Service Technology Series from

Thomas Erl) Oracle PL/SQL Programming Fundamentals: A Tutorial by Examples