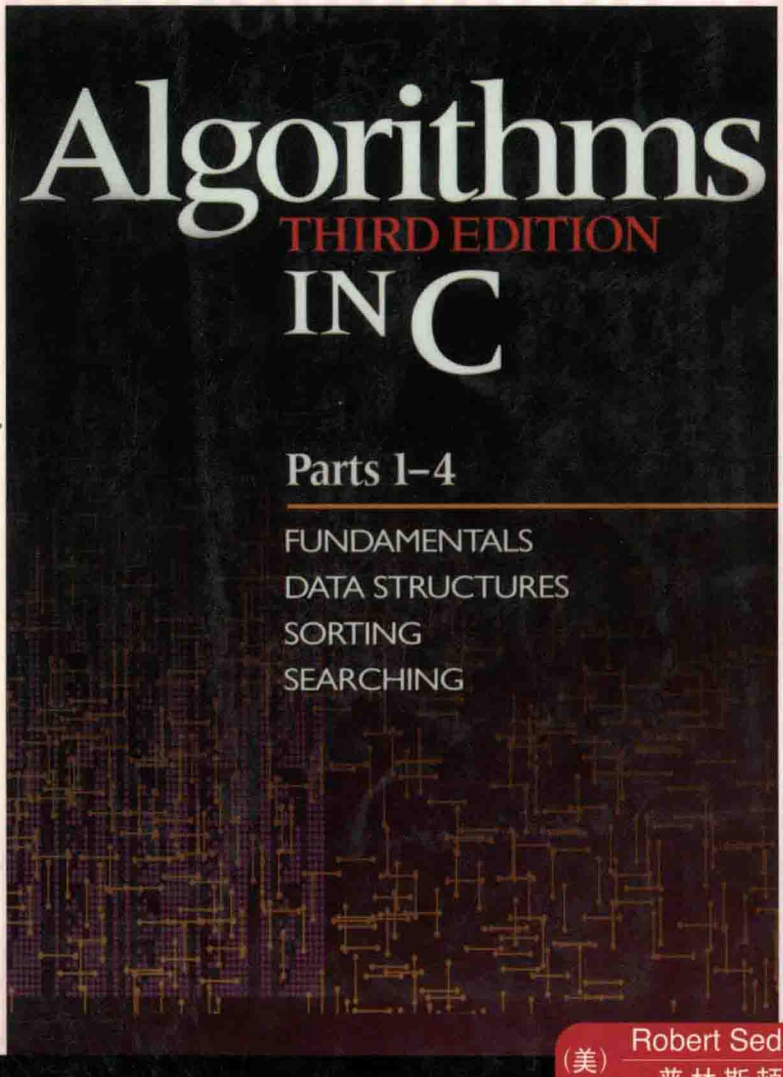
经 典 原 版 书 库

算法：C语言实现

(第1~4部分)

基础知识、数据结构、排序及搜索

(英文版 · 第3版)



(美) Robert Sedgewick 著
普林斯顿大学

机械工业出版社
China Machine Press

算法：C语言实现（第1~4部分）基础知识、数据结构、排序及搜索 （英文版·第3版）

Algorithms in C Parts 1-4: Fundamentals, Data Structures, Sorting, Searching (Third Edition)

对于在数学分析方面不算熟练且需要留意理论算法的普通程序员来说，本书是一本可读性很强的优秀读本。他们应该会从中获益良多。

——Steve Summit，《C Programming FAQs》的作者

Sedgewick有一种真正的天赋，可以用易于理解的方式来解释概念。书中采用了一些易懂的实战程序，其篇幅仅有一页左右，这更是锦上添花。而书中大量采用的图、程序、表格也会极大帮助读者的学习和理解，这使本书更显得与众不同。

——William A. Ward，南亚拉巴马大学

本书是Sedgewick彻底修订和重写的C算法系列的第一本。全书分为四部分，共16章。第一部分“基础知识”（第1~2章）介绍基本算法分析原理。第二部分“数据结构”（第3~5章）讲解算法分析中必须掌握的数据结构知识，主要包括基本数据结构、抽象数据结构、递归和树。第三部分“排序”（第6~11章）按章节顺序分别讨论基本排序方法（如选择排序、插入排序、冒泡排序、希尔排序等）、快速排序方法、归并和归并排序方法、优先队列与堆排序方法、基数排序方法以及特殊目的排序方法，并比较了各种排序方法的性能特征。第四部分“搜索”（第12~16章）在进一步讲解符号表、树等抽象数据类型的基础上，重点讨论哈希方法、基数搜索以及外部搜索方法。

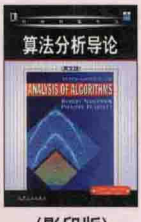
书中提供了用C语言描述的完整算法源程序，并且配有丰富的插图和练习。作者用简洁的实现将理论和实践成功地结合了起来，这些实现均可在真实应用上测试，使得本书自问世以来备受程序员的欢迎。

本书可作为高等院校计算机相关专业算法与数据结构课程的教材和补充读物，也可供自学之用。

本书作者的网站http://www.cs.princeton.edu/~rs/为程序员提供了本书的源代码和勘误表。

作者简介

Robert Sedgewick 拥有斯坦福大学博士学位（导师为Donald E. Knuth），普林斯顿大学计算机科学系教授，Adobe Systems公司董事，曾是Xerox PARC的研究人员，还曾就职于美国国防部防御分析研究所以及INRIA。除本书外，他还与Philippe Flajolet合著了《算法分析导论》一书。



(中文版)
ISBN 7-111-16441-5
定价：38.00元

(影印版)
ISBN 7-111-18606-0
定价：59.00元

PEARSON Addison Wesley www.PearsonEd.com

上架指导：计算机/算法与计算理论/算法

ISBN 7-111-19764-X
9 787111 197645

封面设计：吴 刚

HZ BOOKS 华章图书

For sale and distribution in the People's Republic of China exclusively (except Taiwan, Hong Kong SAR and Macau SAR).
仅限于中华人民共和国境内（不包括中国香港、澳门特别行政区和中国台湾地区）销售发行。

华章网站 http://www.hzbook.com

网上购书：www.china-pub.com

投稿热线：(010) 88379604
购书热线：(010) 68995259, 68995264
读者信箱：hzjsj@hzbook.com

ISBN 7-111-19764-X
定价：69.00元

HZ BOOKS

PEARSON Addison Wesley

算法：C语言实现

（第1～4部分）基础知识、数据结构、排序及搜索

Algorithms in C

Parts 1-4: Fundamentals, Data Structures, Sorting, Searching *(Third Edition)*

（美）Robert Sedgewick 著

出版社
ne Press

经 典 原 版 书 库

算法：C语言实现

(第1~4部分)
基础知识、数据结构、排序及搜索

(英文版 · 第3版)

Algorithms in C

Parts 1-4: Fundamentals, Data Structures, Sorting, Searching

(Third Edition)

(美) Robert Sedgewick
普林斯顿大学 著

机械工业出版社
China Machine Press

本书版权登记号：图字：01-2006-3992

图书在版编目（CIP）数据

算法：C语言实现（第1～4部分）：基础知识、数据结构、排序及搜索（英文版·第3版）/（美）塞奇威克（Sedgewick, R.）著. -北京：机械工业出版社，2006.9

（经典原版书库）

书名原文：Algorithms in C, Parts 1-4: Fundamentals, Data Structures, Sorting, Searching, Third Edition

ISBN 7-111-19764-X

Ⅰ.算…　Ⅱ.塞…　Ⅲ.①电子计算机-算法理论-英文　②C语言-程序设计-英文　Ⅳ.①TP301.6　②TP312

中国版本图书馆CIP数据核字（2006）第096595号

机械工业出版社（北京市西城区百万庄大街22号　邮政编码　100037）

责任编辑：迟振春

北京瑞德印刷有限公司印刷·新华书店北京发行所发行

2006年9月第1版第1次印刷

170mm×242mm·45.25印张

定价：69.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

本社购书热线：（010）68326294

出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭橥了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短、从业人员较少的现状下，美国等发达国家在其计算机科学发展的几十年间积淀的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章图文信息有限公司较早意识到“出版要为教育服务”。自1998年开始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall，Addison-Wesley，McGraw-Hill，Morgan Kaufmann等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Tanenbaum，Stroustrup，Kernighan，Jim Gray等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及庋藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专程为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都步入一个新的阶段。为此，华章公司将加大引进教材的力度，在“华章教育”的总规划之下出版三个系列的计算机教材：除“计算机科学丛书”之外，对影印版的教材，则单独开辟出“经典原版书库”；同时，引进全美通行的教学辅导书“Schaum's Outlines”系列组成“全美经典学习指导系列”。为了保证这三套丛书的权威性，同时也为了更好地为学校和老师们服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔

滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

这三套丛书是响应教育部提出的使用外版教材的号召，为国内高校的计算机及相关专业的教学度身订造的。其中许多教材均已为M. I. T.，Stanford，U.C. Berkeley，C. M. U. 等世界名牌大学所采用。不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程，而且各具特色——有的出自语言设计者之手、有的历经三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下，读者必将在计算机科学的宫殿中由登堂而入室。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证，但我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。教材的出版只是我们的后续服务的起点。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下:

电子邮件: hzjsj@hzbook.com
联系电话: (010) 68995264
联系地址: 北京市西城区百万庄南街1号
邮政编码: 100037

专家指导委员会

（按姓氏笔画顺序）

尤晋元	王　珊	冯博琴	史忠植	史美林
石教英	吕　建	孙玉芳	吴世忠	吴时霖
张立昂	李伟琴	李师贤	李建中	杨冬青
邵维忠	陆丽娜	陆鑫达	陈向群	周伯生
周克定	周傲英	孟小峰	岳丽华	范　明
郑国梁	施伯乐	钟玉琢	唐世渭	袁崇义
高传善	梅　宏	程　旭	程时端	谢希仁
裘宗燕	戴　葵			

To Adam, Andrew, Brett, Robbie,
and especially Linda

Preface

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 basic programming skills and familiarity with computer systems, but before they have taken specialized courses in advanced areas of computer science or computer applications. The book also may be useful for self-study or as a reference for people engaged in the development of computer systems or applications programs, since it contains implementations of useful algorithms and detailed information on these algorithms' performance characteristics. The broad perspective taken makes the book an appropriate introduction to the field.

I have completely rewritten the text for this new edition, and I have added more than a thousand new exercises, more than a hundred new figures, and dozens of new programs. I have also added detailed commentary on all the figures and programs. This new material provides both coverage of new topics and fuller explanations of many of the classic algorithms. A new emphasis on abstract data types throughout the book makes the programs more broadly useful and relevant in modern object-oriented programming environments. People who have read old editions of the book will find a wealth of new information throughout; all readers will find a wealth of pedagogical material that provides effective access to essential concepts.

Due to the large amount of new material, we have split the new edition into two volumes (each about the size of the old edition) of which this is the first. This volume covers fundamental concepts, data structures, sorting algorithms, and searching algorithms; the second volume covers advanced algorithms and applications, building on the basic abstractions and methods developed here. Nearly all the material on fundamentals and data structures in this edition is new.

This book is not just for programmers and computer-science students. Nearly everyone who uses a computer wants it to run faster or to solve larger problems. The algorithms in this book represent a body of knowledge developed over the last 50 years that has become indispensible in the efficient use of the computer, for a broad variety of applications. From N-body simulation problems in physics to genetic-sequencing problems in molecular biology, the basic methods described here have become essential in scientific research; and from database systems to Internet search engines, they have become essential parts of modern software systems. As the scope of computer applications becomes more widespread, so grows the impact of many of the basic methods covered here. The goal of this book is to serve as a resource for students and professionals interested in knowing and making intelligent use of these fundamental algorithms as basic tools for whatever computer application they might undertake.

Scope

The book contains 16 chapters grouped into four major parts: fundamentals, data structures, sorting, and searching. The descriptions here are intended to give readers an understanding of the basic properties of as broad a range of fundamental algorithms as possible. Ingenious methods ranging from binomial queues to patricia tries are described, all related to basic paradigms at the heart of computer science. The second volume consists of four additional parts that cover strings, geometry, graphs, and advanced topics. My primary goal in developing these books has been to bring together the fundamental methods from these diverse areas, to provide access to the best methods known for solving problems by computer.

You will most appreciate the material in this book if you have had one or two previous courses in computer science or have had equivalent programming experience: one course in programming in a high-level language such as C, Java, or C++, and perhaps another course that 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 to fill in gaps in your background are suggested in the text.

Most of the mathematical material supporting the analytic results is self-contained (or is labeled as beyond the scope of this book), so little specific preparation in mathematics is required for the bulk of the book, although mathematical maturity is definitely helpful.

Use in the Curriculum

There is a great deal of flexibility in how the material here can be taught, depending on the taste of the instructor and the preparation of the students. The algorithms described here have found widespread use for years, and represent an essential body of knowledge for both the practicing programmer and the computer-science student. There is sufficient coverage of basic material for the book to be used for a course on data structures, and there is sufficient detail and coverage of advanced material for the book to be used for a course on algorithms. Some instructors may wish to emphasize implementations and practical concerns; others may wish to emphasize analysis and theoretical concepts.

A complete set of slide masters for use in lectures, sample programming assignments, interactive exercises for students, and other course materials may be found via the book's home page.

An elementary course on data structures and algorithms might emphasize the basic data structures in Part 2 and their use in the implementations in Parts 3 and 4. A course on design and analysis of algorithms might emphasize the fundamental material in Part 1 and Chapter 5, then study the ways in which the algorithms in Parts 3 and 4 achieve good asymptotic performance. A course on software engineering might omit the mathematical and advanced algorithmic material, and 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.

Earlier editions of this book have been used in recent years at scores of colleges and universities around the world 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 be expanded upon in later courses on analysis of algorithms, systems programming and

theoretical computer science, while providing the growing group of students from other disciplines with a large set of techniques that these people can immediately put to good use.

The exercises—most of which are new to this edition—fall into several types. Some are intended to test understanding of material in the text, and simply ask readers to work through an example or to apply concepts described in the text. Others involve implementing and putting together the algorithms, or running empirical studies to compare variants of the algorithms and to learn their properties. Still others are a repository for important information at a level of detail that is not appropriate for the text. Reading and thinking about the exercises will pay dividends for every reader.

Algorithms of Practical Use

Anyone wanting to use a computer more effectively can use this book for reference or for self-study. People with programming experience can find information on specific topics throughout the book. To a large extent, you can read the individual chapters in the book independently of the others, although, in some cases, algorithms in one chapter make use of methods from a previous chapter.

The orientation of the book is to study algorithms likely to be of practical use. The book provides information about the tools of the trade to the point that readers can confidently implement, debug, and put to work algorithms to solve a problem or to provide functionality in an application. Full implementations of the methods discussed are included, as are descriptions of the operations of these programs on a consistent set of examples. Because we work with real code, rather than write pseudo-code, the programs can be put to practical use quickly. Program listings are available from the book's home page.

Indeed, one practical application of the algorithms has been to produce the hundreds of figures throughout the book. Many algorithms are brought to light on an intuitive level through the visual dimension provided by these figures.

Characteristics of the algorithms and of the situations in which they might be useful are discussed in detail. Although not emphasized, connections to the analysis of algorithms and theoretical computer science are developed in context. When appropriate, empirical and

analytic results are presented 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 and implementations is synthesized, encapsulated, and discussed throughout the book.

Programming Language

The programming language used for all of the implementations is C. 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 be translated easily to other modern programming languages, since relatively few constructs are unique to C. We use standard C idioms when appropriate, but this book is not intended to be a reference work on C programming.

There are many new programs in this edition, and many of the old ones have been reworked, primarily to make them more readily useful as abstract-data-type implementations. Extensive comparative empirical tests on the programs are discussed throughout the text.

Previous editions of the book have presented basic programs in Pascal, C++, and Modula-3. This code is available through the book home page on the web; code for new programs and code in new languages such as Java will be added as appropriate.

A goal of this book is to present the algorithms in as simple and direct a form as possible. The style is consistent whenever possible, so that programs that are similar look similar. For many of the algorithms in this book, the similarities hold regardless of the language: Quicksort is quicksort (to pick one prominent example), whether expressed in Algol-60, Basic, Fortran, Smalltalk, Ada, Pascal, C, PostScript, Java, or countless other programming languages and environments where it has proved to be an effective sorting method.

We strive for elegant, compact, and portable implementations, but we take the point of view that efficiency matters, so we try to be aware of the performance characteristics of our code at all stages of development. Chapter 1 constitutes a detailed example of this approach to developing efficient C implementations of our algorithms, and sets the stage for the rest of the book.

Acknowledgments

Many people gave me helpful feedback on earlier versions of this book. In particular, hundreds of students at Princeton and Brown have suffered through preliminary drafts over the years. Special thanks are due to Trina Avery and Tom Freeman for their help in producing the first edition; to Janet Incerpi for her creativity and ingenuity in persuading our early and primitive digital computerized typesetting hardware and software to produce the first edition; to Marc Brown for his part in the algorithm visualization research that was the genesis of so many of the figures in the book; and to Dave Hanson for his willingness to answer all of my questions about C. I would also like to thank the many readers who have provided me with detailed comments about various editions, including Guy Almes, Jon Bentley, Marc Brown, Jay Gischer, Allan Heydon, Kennedy Lemke, Udi Manber, Dana Richards, John Reif, M. Rosenfeld, Stephen Seidman, Michael Quinn, and William Ward.

To produce this new edition, I have had the pleasure of working with Peter Gordon and Debbie Lafferty at Addison-Wesley, who have patiently shepherded this project as it has evolved from a standard update to a massive rewrite. It has also been my pleasure to work with several other members of the professional staff at Addison-Wesley. The nature of this project made the book a somewhat unusual challenge for many of them, and I much appreciate their forbearance.

I have gained two new mentors in writing this book, and particularly want to express my appreciation to them. First, Steve Summit carefully checked early versions of the manuscript on a technical level, and provided me with literally thousands of detailed comments, particularly on the programs. Steve clearly understood my goal of providing elegant, efficient, and effective implementations, and his comments not only helped me to provide a measure of consistency across the implementations, but also helped me to improve many of them substantially. Second, Lyn Dupre also provided me with thousands of detailed comments on the manuscript, which were invaluable in helping me not only to correct and avoid grammatical errors, but also—more important—to find a consistent and coherent writing style that helps bind together the daunting mass of technical material here. I am extremely grateful

for the opportunity to learn from Steve and Lyn—their input was vital in the development of this book.

Much of what I have written here I have learned from the teaching and writings of Don Knuth, my advisor at Stanford. Although 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 the scientific footing that makes a work such as this possible. My friend and colleague Philippe Flajolet, who has been a major force in the development of the analysis of algorithms as a mature research area, has had a similar influence on this work.

I am deeply thankful for the support of Princeton University, Brown University, and the Institut National de Recherce en Informatique et Automatique (INRIA), where I did most of the work on the book; and of 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 thank Bill Bowen, Aaron Lemonick, and Neil Rudenstine for their support in building an academic environment at Princeton in which I was able to prepare this book, despite my numerous other responsibilities.

Robert Sedgewick
Marly-le-Roi, France, February, 1983
Princeton, New Jersey, January, 1990
Jamestown, Rhode Island, August, 1997