

## <<算法设计与分析>>

### 图书基本信息

书名：<<算法设计与分析>>

13位ISBN编号：9787508318042

10位ISBN编号：7508318048

出版时间：2003-1

出版时间：中国电力出版社

作者：(美)亚荷

页数：347

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## 前言

The study of algorithms is at the very heart of computer science. In recent years a number of significant advances in the field of algorithms have been made. These advances have ranged from the development of faster algorithms, such as the fast Fourier transform, to the startling discovery of certain natural problems for which all algorithms are inefficient. These results have kindled considerable interest in the study of algorithms, and the area of algorithm design and analysis has blossomed into a field of intense interest. The intent of this book is to bring together the fundamental results in this area, so the unifying principles and underlying concepts of algorithm design may more easily be taught.

**THE SCOPE OF THE BOOK** To analyze the performance of an algorithm some model of a computer is necessary. Our book begins by formulating several computer models which are simple enough to establish analytical results but which at the same time accurately reflect the salient features of real machines. These models include the random access register machine, the random access stored program machine, and some specialized variants of these. The Turing machine is introduced in order to prove the exponential lower bounds on efficiency in Chapters 10 and 11. Since the trend in program design is away from machine language, a high-level language called Pidgin ALGOL is introduced as the main vehicle for describing algorithms. The complexity of a Pidgin ALGOL program is related to the machine models. The second chapter introduces basic data structures and programming techniques often used in efficient algorithms. It covers the use of lists, pushdown stores, queues, trees, and graphs. Detailed explanations of recursion, divide-and-conquer, and dynamic programming are given, along with examples of their use. Chapters 3 to 9 provide a sampling of the diverse areas to which the fundamental techniques of Chapter 2 can be applied. Our emphasis in these chapters is on developing algorithms that are asymptotically the most efficient known. Because of this emphasis, some of the algorithms presented are suitable only for inputs whose size is much larger than what is currently encountered in practice. This is particularly true of some of the matrix multiplication algorithms in Chapter 6, the Schonhage-Strassen integer-multiplication algorithm of Chapter 7, and some of the polynomial and integer algorithms of Chapter 8. On the other hand, most of the sorting algorithms of Chapter 3, the searching algorithms of Chapter 4, the graph algorithms of Chapter 5, the fast Fourier transform of Chapter 7, and the string-matching algorithms of Chapter 9 are widely used, since the sizes of inputs for which these algorithms are efficient are sufficiently small to be encountered in many practical situations. Chapters 10 through 12 discuss lower bounds on computational complexity. The inherent computational difficulty of a problem is of universal interest, both to program design and to an understanding of the nature of computation. In Chapter 10 an important class of problems, the NP-complete problems, is studied. All problems in this class are equivalent in computational difficulty, in that if one problem in the class has an efficient (polynomial time-bounded) solution, then all problems in the class have efficient solutions. Since this class of problems contains many practically important and wellstudied problems, such as the integer-programming problem and the traveling salesman problem, there is good reason to suspect that no problem in this class can be solved efficiently. Thus, if a program designer knows that the problem for which he is trying to find an efficient algorithm is in this class, then he may very well be content to try heuristic approaches to the problem. In spite of the overwhelming empirical evidence to the contrary, it is still an open question whether NP-complete problems admit of efficient solutions. In Chapter 11 certain problems are defined for which we can actually prove that no efficient algorithms exist. The material in Chapters 10 and 11 draws heavily on the concept of Turing machines introduced in Sections 1.6 and 1.7. In the final chapter of the book we relate concepts of computational difficulty to notions of linear independence in vector spaces. The material in this chapter provides techniques for proving lower bounds for much simpler problems than those considered in Chapters 10 and 11.

**THE USE OF THE BOOK** This book is intended as a first course in the design and analysis of algorithms. The emphasis is on ideas and ease of understanding rather than implementation details or programming tricks. Informal, intuitive explanations are often used in place of long tedious proofs. The book is self-contained and assumes no specific background in mathematics or programming languages. However, a certain amount of maturity in being able to handle mathematical concepts is desirable, as is some exposure to a higher-level programming language such as

FORTRAN or ALGOL. Some knowledge of linear algebra is needed for a full understanding of Chapters 6, 7, 8, and 12. This book has been used in graduate and undergraduate courses in algorithm design. In a one-semester course most of Chapters 1-5 and 9-10 were covered, along with a smattering of topics from the remaining chapters. In introductory courses the emphasis was on material from Chapters 1-5, but Sections 1.6, 1.7, 4.13, 5.11, and Theorem 4.5 were generally not covered. The book can also be used in more advanced courses emphasizing the theory of algorithms. Chapters 6-12 could serve as the foundation for such a course. Numerous exercises have been provided at the end of each chapter to provide an instructor with a wide range of homework problems. The exercises are graded according to difficulty. Exercises with no stars are suitable for introductory courses, singly starred exercises for more advanced courses, and doubly starred exercises for advanced graduate courses. The bibliographic notes at the end of every chapter attempt to point to a published source for each of the algorithms and results contained in the text and the exercises.

**ACKNOWLEDGMENTS** The material in this book has been derived from class notes for courses taught by the authors at Cornell, Princeton, and Stevens Institute of Technology. The authors would like to thank the many people who have critically read various portions of the manuscript and offered many helpful suggestions. In particular we would like to thank Kellogg Booth, Stan Brown, Steve Chen, Allen Cypher, Arch Davis, Mike Fischer, Hania Gajewska, Mike Garey, Udai Gupta, Mike Harrison, Matt Hecht, Harry Hunt, Dave Johnson, Marc Kaplan, Don Johnson, Steve Johnson, Brian Kernighan, Don Knuth, Richard Ladner, Anita LaSalle, Doug McIlroy, Albert Meyer, Christos Papadimitriou, Bill Plauser, John Savage, Howard Siegel, Ken Steiglitz, Larry Stockmeyer, Tom Szymanski, and Theodore Yen. Special thanks go to Gemma Carnevale, Pauline Cameron, Hannah Kresse, Edith Purser, and Ruth Suzuki for their cheerful and careful typing of the manuscript. The authors are also grateful to Bell Laboratories and Cornell, Princeton, and the University of California at Berkeley for providing facilities for the preparation of the manuscript.

## <<算法设计与分析>>

### 内容概要

跳过高深的理论和数学公式，直击现实世界中的数字设计和测试工作。

了解使用策略，满足对品质、可靠性及成本控制的业务需求，使你能在典型的高压力环境下按期完成任务。

本书将帮助你优化设计，在工程上权衡某些资源（如硅片面积、工作频率和功率消耗）间的关系；在整体上实现的测试成本、面市时间和量产时间的平衡。

此外，还将通过特定的自动化测试仿真生成工具提升你的效率。

书中包括的索引使你能够根据自己的需要，直接阅读你所关注的内容。

主要内容包括：

- 设计核心，关注嵌入核心和嵌入存储器

- 系统集成和超大规模集成电路的设计问题

- AC扫描、正常速度扫描和嵌入式可测试性设计

- 内建、自测试、含内存BIST、逻辑BIST及扫描BIST

- 虚拟测试套接字和隔离测试

- 重用设计，包括重用向量和重用核心

- 用VSIA和IEEE P1500标准处理测试问题

- 书中穿插的整幅图解直接来自作者的教学材料。

通过为书中的每一部分列出流程图、工程图表和内容摘要，使得读者能够更快更容易地学习和查找。

本书是与设计和测试工作相关的工程师和管理员所必备的资料书籍。

## <<算法设计与分析>>

### 作者简介

Alfred V.Aho,朗讯科技贝尔实验室的研发副总裁。

Aho获得了加拿大多伦多大学的学士学位和美国普林斯顿大学的硕士和博士学位。

Aho是美国国家工程院院士, ACM、IEEE、AAAS的Fellow, 并且担任ACM自动控制与可计算性理论特别兴趣组的副主席和美国国家科学基金会计算机与信息技术顾问委员会主席。

John E.Hopcroft,美国康乃尔大学的教授、工程院院长他获得了美国斯坦福大学的硕士和博士学位Hopcroft是美国国家工程院院士, ACM、IEEE、AAAS的Fellow, 并且获得了1986年度ACM图灵奖, 他还是多个国际著名刊物的主编。

Jeffrey D.Ullman,美国斯坦福大学计算机科学系的教授他获得了美国哥伦比亚大学的学士学位和普林斯顿大学的博士学位、Ullman是美国国家工程院院士, ACM的Fellow他获得1998年度ACM Karl

V.Karlstrom的杰出教育成就奖和2000年度的Knuth奖金。

## &lt;&lt;算法设计与分析&gt;&gt;

## 书籍目录

1 Models of Computation  
 1.1 Algorithms and their complexity  
 1.2 Random access machines  
 1.3 Computational complexity of RAM programs  
 1.4 A stored program model  
 1.5 Abstractions of the RAM  
 1.6 A primitive model of computation: the Turing machine  
 1.7 Relationship between the Turing machine and RAM models  
 1.8 Pidgin ALGOL—a high-level language  
 2 Design of Efficient Algorithms  
 2.1 Data structures: lists, queues, and stacks  
 2.2 Set representations  
 2.3 Graphs  
 2.4 Trees  
 2.5 Recursion  
 2.6 Divide-and-conquer  
 2.7 Balancing  
 2.8 Dynamic programming  
 2.9 Epilogue  
 3 Sorting and Order Statistics  
 3.1 The sorting problem  
 3.2 Radix sorting  
 3.3 Sorting by comparisons  
 3.4 Heapsort—an  $O(n \log n)$  comparison sort  
 3.5 Quicksort—an  $O(n \log n)$  expected time sort  
 3.6 Order statistics  
 3.7 Expected time for order statistics  
 4 Data Structures for Set Manipulation Problems  
 4.1 Fundamental operations on sets  
 4.2 Hashing  
 4.3 Binary search  
 4.4 Binary search trees  
 4.5 Optimal binary search trees  
 4.6 A simple disjoint-set union algorithm  
 4.7 Tree structures for the UNION-FIND problem  
 4.8 Applications and extensions of the UNION-FIND algorithm  
 4.9 Balanced tree schemes  
 4.10 Dictionaries and priority queues  
 4.11 Mergeable heaps  
 4.12 Concatenable queues  
 4.13 Partitioning  
 4.14 Chapter summary  
 5 Algorithms on Graphs  
 5.1 Minimum-cost spanning trees  
 5.2 Depth-first search  
 5.3 Biconnectivity  
 5.4 Depth-first search of a directed graph  
 5.5 Strong connectivity  
 5.6 Path-finding problems  
 5.7 A transitive closure algorithm  
 5.8 A shortest-path algorithm  
 5.9 Path problems and matrix multiplication  
 5.10 Single-source problems  
 5.11 Dominators in a directed acyclic graph: putting the concepts together  
 6 Matrix Multiplication and Related Operations  
 6.1 Basics  
 6.2 Strassen's matrix-multiplication algorithm  
 6.3 Inversion of matrices  
 6.4 LU decomposition of matrices  
 6.5 Applications of LUP decomposition  
 6.6 Boolean matrix multiplication  
 7 The Fast Fourier Transform and its Applications  
 7.1 The discrete Fourier transform and its inverse  
 7.2 The fast Fourier transform algorithm  
 7.3 The FFT using bit operations  
 7.4 Products of polynomials  
 7.5 The Schonhage-Strassen integer-multiplication algorithm  
 8 Integer and Polynomial Arithmetic  
 8.1 The similarity between integers and polynomials  
 8.2 Integer multiplication and division  
 8.3 Polynomial multiplication and division  
 8.4 Modular arithmetic  
 8.5 Modular polynomial arithmetic and polynomial evaluation  
 8.6 Chinese remaindering  
 8.7 Chinese remaindering and interpolation of polynomials  
 8.8 Greatest common divisors and Euclid's algorithm  
 8.9 An asymptotically fast algorithm for polynomial GCD's  
 8.10 Integer GCD's  
 8.11 Chinese remaindering revisited  
 8.12 Sparse polynomials  
 9 Pattern-Matching Algorithms  
 9.1 Finite automata and regular expressions  
 9.2 Recognition of regular expression patterns  
 9.3 Recognition of substrings  
 9.4 Two-way deterministic pushdown automata  
 9.5 Position trees and substring identifiers  
 10 NP-Complete Problems  
 10.1 Nondeterministic Turing machines  
 10.2 The classes P and NP  
 10.3 Languages and problems  
 10.4 NP-completeness of the satisfiability problem  
 10.5 Additional NP-complete problems  
 10.6 Polynomial-space-bounded problems  
 11 Some Provably Intractable Problems  
 11.1 Complexity hierarchies  
 11.2 The space hierarchy for deterministic Turing machines  
 11.3 A problem requiring exponential time and space  
 11.4 A nonelementary problem  
 12 Lower Bounds on Numbers of Arithmetic Operations  
 12.1 Fields  
 12.2 Straight-line code revisited  
 12.3 A matrix formulation of problems  
 12.4 A row-oriented lower bound on multiplications  
 12.5 A column-oriented lower bound on multiplications  
 12.6 A row-and-column-oriented bound on multiplications  
 12.7 Preconditioning  
 Bibliography  
 Index

<<算法设计与分析>>

章节摘录

插图：

## <<算法设计与分析>>

### 编辑推荐

《算法设计与分析(影印版)》适用于本科生和研究生的算法设计课程每章后面提供了大量的练习练习根据难度进行了分级，读者可以根据不同的需要选择。



<<算法设计与分析>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>