HZ BOOKS

PEARSON Addison Wesley

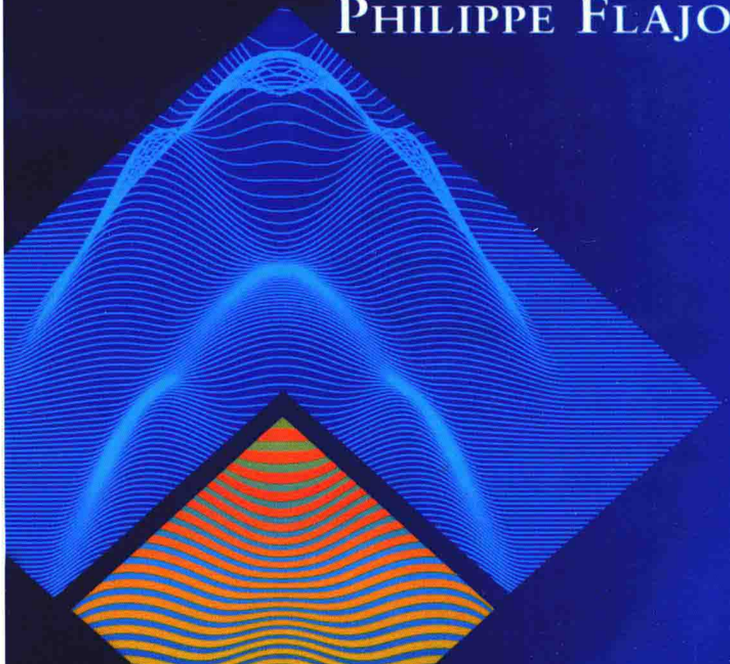
经 典 原 版 书 库

算法分析导论

(英文版)

AN INTRODUCTION TO THE ANALYSIS OF ALGORITHMS

ROBERT SEDGEWICK
PHILIPPE FLAJOLET



(美) Robert Sedgewick
Philippe Flajolet 著

机械工业出版社
China Machine Press

经典原版书库

算法分析导论

(英文版)

An Introduction to the Analysis of Algorithms

(美) Robert Sedgewick
Philippe Flajolet 著

机械工业出版社
China Machine Press

本书版权登记号：图字：01-2006-1102

图书在版编目（CIP）数据

算法分析导论（英文版）/（美）塞奇威克（Sedgewick, R.）等著. －北京：机械工业出版社，2006.4

（经典原版书库）

书名原文：An Introduction to the Analysis of Algorithms

ISBN 7-111-18606-0

Ⅰ. 算…　Ⅱ. 塞…　Ⅲ. 算法分析－英文　Ⅳ. TP311

中国版本图书馆CIP数据核字（2006）第017264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

北京京北制版厂印刷 · 新华书店北京发行所发行

2006年4月第 1 版第 1 次印刷

718mm × 1020mm 1/16 · 31.75印张

定价：59.00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，还揭櫫了学术的源变，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

FOREWORD

PEOPLE who analyze algorithms have double happiness. First of all they experience the sheer beauty of elegant mathematical patterns that surround elegant computational procedures. Then they receive a practical payoff when their theories make it possible to get other jobs done more quickly and more economically.

Mathematical models have been a crucial inspiration for all scientific activity, even though they are only approximate idealizations of real-world phenomena. Inside a computer, such models are more relevant than ever before, because computer programs create artificial worlds in which mathematical models often apply precisely. I think that's why I got hooked on analysis of algorithms when I was a graduate student, and why the subject has been my main life's work ever since.

Until recently, however, analysis of algorithms has largely remained the preserve of graduate students and post-graduate researchers. Its concepts are not really esoteric or difficult, but they are relatively new, so it has taken awhile to sort out the best ways of learning them and using them.

Now, after more than 30 years of development, algorithmic analysis has matured to the point where it is ready to take its place in the standard computer science curriculum. The appearance of this long-awaited textbook by Sedgewick and Flajolet is therefore most welcome. Its authors are not only worldwide leaders of the field, they also are masters of exposition. I am sure that every serious computer scientist will find this book rewarding in many ways.

D. E. Knuth

PREFACE

THIS book is intended to be a thorough overview of the primary techniques used in the mathematical analysis of algorithms. The material covered draws from classical mathematical topics, including discrete mathematics, elementary real analysis, and combinatorics, as well as from classical computer science topics, including algorithms and data structures. The focus is on "average-case" or "probabilistic" analysis, though the basic mathematical tools required for "worst-case" or "complexity" analysis are covered as well.

We assume that the reader has some familiarity with basic concepts in both computer science and real analysis. In a nutshell, the reader should be able to both write programs and prove theorems; otherwise, the book is intended to be self-contained. Ample references to preparatory material in the literature are also provided. A planned companion volume will cover more advanced techniques. Together, the books are intended to cover the main techniques and to provide access to the growing research literature on the analysis of algorithms.

The book is meant to be used as a textbook in a junior- or senior-level course on "Mathematical Analysis of Algorithms." It might also be useful in a course in discrete mathematics for computer scientists, since it covers basic techniques in discrete mathematics as well as combinatorics and basic properties of important discrete structures within a familiar context for computer science students. It is traditional to have somewhat broader coverage in such courses, but many instructors may find the approach here a useful way to engage students in a substantial portion of the material. The book also can be used to introduce students in mathematics and applied mathematics to principles from computer science related to algorithms and data structures.

Supplemented by papers from the literature, the book can serve as the basis for an introductory graduate course on the analysis of algorithms, or as a reference or basis for self-study by researchers in mathematics or computer science who want access to the literature in this field. It also might be of use to students and researchers in combinatorics and discrete mathematics, as a source of applications and techniques.

Despite the large literature on the mathematical analysis of algorithms, basic information on methods and models in widespread use has not been directly accessible to students and researchers in the field. This book aims to address this situation, bringing together a body of material intended to provide the reader with both an appreciation for the challenges of the field and the requisite background for learning the advanced tools being developed to meet these challenges.

Preparation. Mathematical maturity equivalent to one or two years' study at the college level is assumed. Basic courses in combinatorics and discrete mathematics may provide useful background (and may overlap with some material in the book), as would courses in real analysis, numerical methods, or elementary number theory. We draw on all of these areas, but summarize the necessary material here, with reference to standard texts for people who want more information.

Programming experience equivalent to one or two semesters' study at the college level, including elementary data structures, is assumed. We do not dwell on programming and implementation issues, but algorithms and data structures are the central object of our studies. Again, our treatment is complete in the sense that we summarize basic information, with reference to standard texts and primary sources.

Access to a computer system for mathematical manipulation, such as MAPLE or Mathematica, is highly recommended. These systems can relieve one from tedious calculations when checking material in the text or solving exercises.

Related books. Related texts include *The Art of Computer Programming* by Knuth; *Handbook of Algorithms and Data Structure* by Gonnet and Baeza-Yates; *Algorithms* by Sedgewick; *Concrete Mathematics* by Graham, Knuth and Patashnik; and *Introduction to Algorithms* by Cormen, Leiserson, and Rivest. This book could be considered supplementary to each of these, as examined below, in turn.

In spirit, this book is closest to the pioneering books by Knuth. Our focus is on mathematical techniques of analysis, though, whereas Knuth's books are broad and encyclopedic in scope, with properties of algorithms playing a primary role and methods of analysis a secondary role. This book can serve as basic preparation for the advanced results covered and referred to in Knuth's books.

We also cover approaches and results in the analysis of algorithms that have been developed since publication of Knuth's books. Gonnet and

Baeza-Yates's *Handbook* is a thorough survey of such results, including a comprehensive bibliography; it primarily presents results with reference to derivations in the literature. Again, our book provides the basic preparation for access to this literature.

We also strive to keep the focus on covering algorithms of fundamental importance and interest, such as those described in Sedgewick's *Algorithms*, whereas Graham, Knuth, and Patashnik's *Concrete Mathematics* focuses almost entirely on mathematical techniques. This book is intended to be a link between the basic mathematical techniques discussed in Graham, Knuth, and Patashnik and the basic algorithms covered in Sedgewick.

Cormen, Leiserson, and Rivest's *Introduction to Algorithms* is representative of a number of books that provide access to the research literature on "design and analysis" of algorithms, which is normally based on rough worst-case estimates of performance. When more precise results are desired (presumably for the most important methods), more sophisticated models and mathematical tools are required. Cormen, Leiserson, and Rivest focus on *design* of algorithms (usually with the goal of bounding worst-case performance), with analytic results used to help guide the design. Our book, on the other hand, is supplementary to theirs and similar books in that we focus on the *analysis* of algorithms, especially on techniques that can be used to develop detailed results that could be used to predict performance. In this process, we also consider relationships to various classical mathematical tools. Chapter 1 is devoted entirely to developing this context.

This book also lays the groundwork for a companion volume, *Analytic Combinatorics*, a general treatment that places the material here in a broader perspective and develops advanced methods and models that can serve as the basis for new research, not only in average-case analysis of algorithms but also in combinatorics. A higher level of mathematical maturity is assumed for that volume, perhaps at the senior or beginning graduate student level. Of course, careful study of this book is adequate preparation. It certainly has been our goal to make it sufficiently interesting that some readers will be inspired to tackle more advanced material!

How to use this book. Readers of this book are likely to have rather diverse backgrounds in discrete mathematics and computer science. With this in mind, it is useful to be aware the basic structure of book: eight chapters in all, an introductory chapter followed by three chapters emphasizing mathematical methods, then four chapters emphasizing applications in the analysis of algorithms:

INTRODUCTION
- 1. ANALYSIS OF ALGORITHMS

DISCRETE MATHEMATICAL METHODS
- 2. RECURRENCES
- 3. GENERATING FUNCTIONS
- 4. ASYMPTOTIC ANALYSIS

ALGORITHMS AND COMBINATORIAL STRUCTURES
- 5. TREES
- 6. PERMUTATIONS
- 7. STRINGS AND TRIES
- 8. WORDS AND MAPS

Chapter 1 puts the material in the book into perspective, and will help all readers understand the basic objectives of the book and the role of the remaining chapters in meeting those objectives. Chapters 2 through 4 are more oriented towards mathematics, as they cover methods from discrete mathematics, primarily focused on developing basic concepts and techniques. Chapters 5 through 8 are more oriented towards computer science, as they cover properties of combinatorial structures, their relationships to fundamental algorithms, and analytic results.

Though the book is intended to be self-contained, differences in emphasis are appropriate in teaching the material, depending on the background and experience of students and instructor. One approach, more mathematically oriented, would be to emphasize the theorems and proofs in the first part of the book, with applications drawn from Chapters 5 through 8. Another approach, more oriented towards computer science, would be to briefly cover the major mathematical tools in Chapters 2 through 4 and emphasize the algorithmic material in the second half of the book. But our primary intention is that most students should be able to learn new material from both mathematics and computer science in an interesting context by working carefully all the way through the book.

Students with a strong computer science background are likely to have seen many of the algorithms and data structures from the second half of the book but not much of the mathematical material at the beginning;

students with a strong background in mathematics are likely to find the mathematical material familiar but perhaps not the algorithms and data structures. A course covering all of the material in the book could help either group of students fill in gaps in their background while building upon knowledge they already have.

Supplementing the text are lists of references (at the end of each chapter) and several hundred exercises (interspersed throughout the text), to encourage readers to consider the material in the text in more depth and to examine original sources.

Our experience in teaching this material has shown that there are numerous opportunities for instructors to supplement lecture and reading material with computation-based laboratories and homework assignments. The material covered here is an ideal framework for students to develop expertise in a symbolic manipulation system such as Mathematica or MAPLE. Also, the experience of validating the mathematical studies by comparing them against empirical studies can be very valuable for many students.

Acknowledgments. We are very grateful to INRIA, Princeton University, and the National Science Foundation, which have provided the primary support for us to work on this book. Other support has been provided by Brown University, European Community (Alcom Project), Institute for Defense Analyses, Ministère de la Recherche et de la Technologie, Stanford University, Université Libre de Bruxelles, and Xerox Palo Alto Research Center. This book has been many years in the making, so a comprehensive list of people and organizations that have contributed support would be prohibitively long, and we apologize for any omissions.

Don Knuth's influence on our work has been extremely important, as is obvious from the text.

Students in Princeton, Paris, and Providence have provided helpful feedback in courses taught from early versions of this material over the years. We would also like to thank Philippe Dumas; Mordecai Golin and his students San Kuen Chan, Ka Po Lam, Ngok Hing Leung, Derek Ka-Cheong Lueng, and King Shan Lui; and three anonymous reviewers for their detailed comments on the manuscript.

Corfu R. S.
September 1995 Ph. F.

NOTATION

$\lfloor x \rfloor$ *floor function*
largest integer less than or equal to x

$\lceil x \rceil$ *ceiling function*
smallest integer greater than or equal to x

$\{x\}$ *fractional part*
$x - \lfloor x \rfloor$

$\lg N$ *binary logarithm*
$\log_2 N$

$\ln N$ *natural logarithm*
$\log_e N$

$\binom{n}{k}$ *binomial coefficient*
number of ways to choose k out of n items

$\left[{n \atop k}\right]$ *Stirling number of the first kind*
number of permutations of n elements that have k cycles

$\left\{{n \atop k}\right\}$ *Stirling number of the second kind*
number of ways to partition n elements into k nonempty subsets

ϕ *golden ratio*
$(1 + \sqrt{5})/2 = 1.61803\cdots$

γ *Euler's constant*
$.57721\cdots$

σ *Stirling's constant*
$\sqrt{2\pi} = 2.50662\cdots$

TABLE OF CONTENTS

CHAPTER ONE

ANALYSIS OF ALGORITHMS

MATHEMATICAL studies of the properties of computer algorithms have spanned a broad spectrum, from general complexity studies to specific analytic results. In this chapter, we discuss representative examples of various approaches to studying algorithms. At the same time, these examples allow us to present various classical results from a fundamental and representative problem domain: the study of sorting algorithms.

First, we will consider the general motivations for algorithmic analysis, and relationships among various approaches to studying performance characteristics of algorithms. Next, we discuss computational complexity and consider as an example Mergesort, an "optimal" algorithm for sorting. Following that, we will examine the major components of a full analysis for a sorting algorithm of fundamental practical importance, Quicksort. This includes the study of various improvements to the basic Quicksort algorithm, as well as some examples illustrating how the analysis can help one adjust parameters to improve performance.

This chapter is intended to place our field of study into context among related fields and to set the stage for the rest of the book. In Chapters 2 through 4, our plan is to introduce the basic mathematical concepts needed for the analysis of fundamental algorithms. In Chapters 5 through 8, we consider basic combinatorial properties of fundamental algorithms and data structures. Since there is a close relationship between fundamental methods used in computer science and classical mathematical analysis, we simultaneously consider some introductory material from both areas in this book.

1.1 Why Analyze an Algorithm? There are several answers to this basic question, depending on context: the intended use of the algorithm, the importance of the algorithm in relationship to others from both practical and theoretical standpoints, and the difficulty of analysis and accuracy of the answer required.

The most straightforward reason for analyzing an algorithm is to discover its characteristics in order to evaluate its suitability for various applications or compare it with other algorithms for the same application.