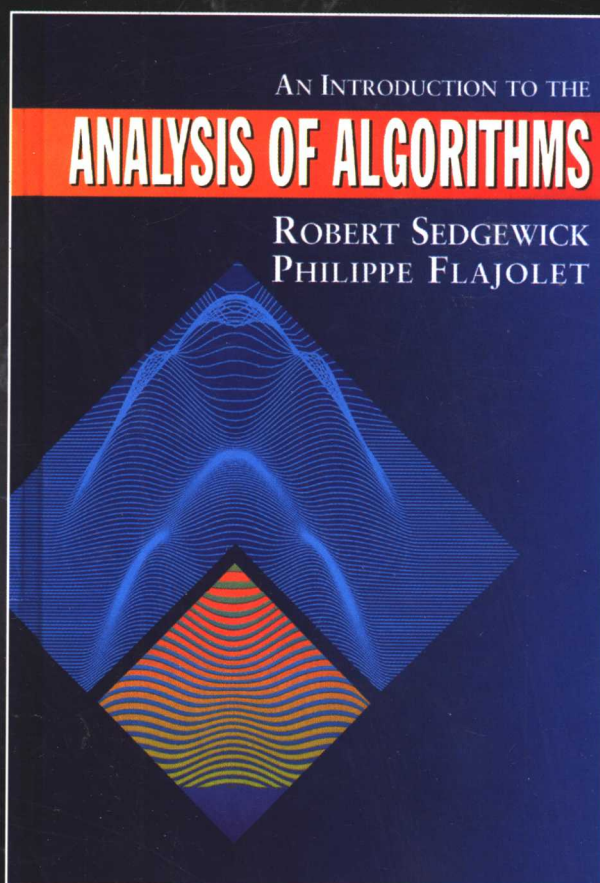





计 算 机 科 学 丛 书

算法分析导论

(美) Robert Sedgewick (法) Philippe Flajolet 著 冯舜玺 李学武 裴伟东 等译



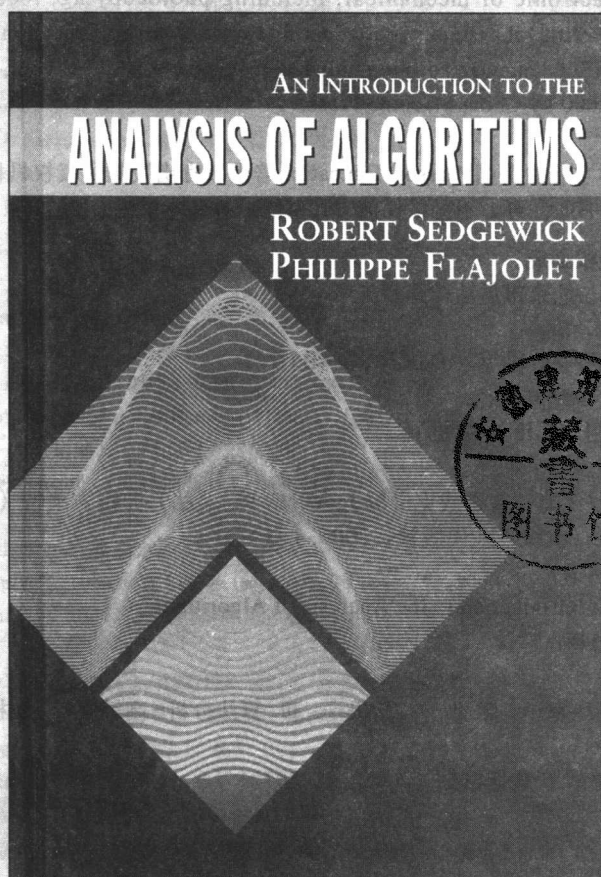
An Introduction to
the Analysis of Algorithms

 机械工业出版社
China Machine Press

计 算 机 科 学 丛 书

算法分析导论

(美) Robert Sedgewick (法) Philippe Flajolet 著 冯舜玺 李学武 裴伟东 等译



**An Introduction to
the Analysis of Algorithms**



机械工业出版社
China Machine Press

本书阐述了用于算法数学分析的主要方法,所涉及的材料来自经典数学课题,包括离散数学、初等实分析、组合数学,以及来自经典的计算机科学课题,包括算法和数据结构。本书内容集中覆盖基础、重要和有趣的算法,前面侧重数学,后面集中讨论算法分析的应用,重点是算法分析的数学方法。每章包含大量习题以及参考文献,使读者可以更深入地理解书中的内容。

本书适合作为高等院校数学、计算机科学以及相关专业的本科生和研究生的教材,也可供相关技术人员参考。

Authorized translation from the English language edition entitled *An introduction to the analysis of algorithms* by Robert Sedgewick, Philippe Flajolet, published by Pearson Education, Inc, publishing as Addison-Wesley (ISBN0-201-40009-X), Copyright © 1999 by Addison-Wesley Publishing Company, Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage retrieval system, without permission of Pearson Education, Inc.

Chinese simplified language edition published by China Machine Press.

Copyright © 2000 by China Machine Press.

本书中文简体字版由美国Pearson Education培生教育出版集团授权机械工业出版社独家出版。未经出版者书面许可,不得以任何方式复制或抄袭本书内容。

版权所有,侵权必究。

本书法律顾问 北京市展达律师事务所

本书版权登记号:图字:01-2003-4919

图书在版编目(CIP)数据

算法分析导论/(美)塞奇威克(Sedgewick, R.)(法)弗拉吉莱特(Flajolet, P)著;冯舜玺等译.-北京:机械工业出版社,2006.4

(计算机科学丛书)

书名原文:An Introduction to the Analysis of Algorithms

ISBN 7-111-16441-5

I.算… II.①塞… ②弗… ③冯… III.算法分析-高等学校-教材 IV.TP311

中国版本图书馆CIP数据核字(2005)第030721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:朱起飞

北京中兴印刷有限公司印刷·新华书店北京发行所发行

2006年4月第1版第1次印刷

787mm×1092mm 1/16·20.5印张

定价:38.00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中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，不仅肇划了研究的范畴，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、研究及度藏。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

石教英

张立昂

邵维忠

周克定

郑国梁

高传善

裘宗燕

王 珊

吕 建

李伟琴

陆丽娜

周傲英

施伯乐

梅 宏

戴 葵

冯博琴

孙玉芳

李师贤

陆鑫达

孟小峰

钟玉琢

程 旭

史忠植

吴世忠

李建中

陈向群

岳丽华

唐世渭

程时端

史美林

吴时霖

杨冬青

周伯生

范 明

袁崇义

谢希仁

译者序

算法分析一般包括两种不同的方法。第一种方法研究确定最坏情形的性能，有时称之为计算复杂性。当面对一种新的算法时，算法分析能粗略地分析：新算法能够有多好？它比其他算法性能大致好多少？就算法分析来说，这种复杂性分析由于抛弃一些次要因素而丧失不少的精度，它提供不了算法具体实现的性能特征以及与其他算法具体比较所需要的足够信息。不过，我们可以把它看成是形成更精练、更准确分析过程的第一步。第二种方法通过确定最佳情形、最坏情形以及平均情形的性能来精确地刻画算法的性能，在需要的时候，能够通过可以精化的方法准确地预测特定算法的性能特征，其中最常见的是对时间和空间这样一些基本资源的消耗，尤其是时间。算法分析表示整个分析过程，它提供任意精度的解决方案。

本书是对算法的数学分析中主要方法的综述，涉及离散数学、数学分析、组合数学以及数据结构和算法。由于重点在于平均情形的分析，因此必然要涉及模型、分布、矩阵以及概率论和数理统计中的其他内容。

经过近几十年的发展，算法分析已经十分成熟，能够在标准的计算机课程中占有一席之地，本书为那些徘徊在该领域之外却苦于不得其门而入的新手指出通往研究之门的捷径。为本科生开设这门课程可以有少数内容现用现补，研究生在先修有关数学和计算机的相应知识的基础上，可以将本书用于算法分析的导论性课程。数学和计算机领域中不同专业的学生，选修本书可以各有侧重，作者在前言中提出了指导性的建议。

本书语言简炼，推导紧凑，能够很好地锻炼我们的独立阅读能力。在验证书中的结论和求解练习的结果时，若能使用像Matlab、Mathematica或MAPLE这样的计算机软件，则可帮助我们摆脱繁琐的演算，节省宝贵的时间。

本书的另一大特点是课文中以及各章末尾所列出的参考文献，对于有心深造的读者，这是十分宝贵的财富。

全书共有8章内容。第3章由李学武翻译，第4章、第6章、第7章和第8章由裴伟东翻译，其余工作由冯舜玺完成。因时间和水平所限，译文难免存在一定的问题，恳切期望广大读者批评指正。

译者

2006年3月

序

分析算法的人享有双重的幸福。首先，他们能够体验到优雅数学模式纯粹的美，这种模式存在于优美的计算过程之中。其次，当他们的理论使得其他工作能够做得更快、更经济时，他们能够得到实际的褒奖。

数学模型始终是所有科学活动至关重要的灵感，尽管它们只是真实世界现象的近似理想化。在计算机内部，这种模型比以前更优美，因为计算机程序创建了人造世界。在这样的世界中，数学模型常常得以精确地使用。我想，这就是为什么当我还是研究生时就沉迷于算法分析，以及为什么这个课题迄今为止始终是我一生主要工作的原因。

然而，直到最近，算法分析在很大程度上依然是研究生和研究人员的禁区。它的概念确实不深奥也不困难，但是它们相对新颖，因此，挑选出学习它们和使用它们的最佳方法需要时间。

如今，经过30多年的发展，算法分析已经成熟到这样一种程度，即它已经为其在标准的计算机课程中占有一席之地做好了准备。因此，我们期盼已久的这部Sedgewick和Flajolet的著作极受欢迎。本书作者不仅是这个领域在世界范围内的领袖，而且还是理论阐述的大师。我敢肯定，每一位严肃的计算机科学家都将发现本书在许多方面颇富启发性。

D. E. Knuth

前 言

本书是对算法数学分析中主要方法的综述。所涉及的材料来自经典的数学课题，包括离散数学、初等实分析、组合数学，以及来自经典的计算机科学课题，包括算法和数据结构。重点在于“平均情形”或“概率”分析，不过，也包括“最坏情形”和“复杂性”分析所需要的基本数学工具。

我们假设读者熟悉计算机科学和实分析中的基本概念。概括地说，读者应该既能编写程序，又能够证明定理；在其他方面，本书是自包容的。书中还在文献里提供了丰富的预备性材料。我们计划中的本书姐妹篇将讨论更为先进的方法。这两部著作总共涵盖那些主要的方法和技巧，并提供通向算法分析日益增长的文献的捷径。

本书计划作为算法数学分析初级和高级课程的教科书。它也可以在计算机科学的离散数学课程中使用，因为它包含了离散数学和组合数学中的基本方法以及计算机科学学生所熟悉的重要离散结构的基本性质。传统上，在这些课程中知识范围要广一些，但是许多教师可能发现这里的做法是使学生从事实质性部分学习的有用方式。本书还可以作为数学和应用数学的学生了解与算法和数据结构相关的计算机科学原理的教材。

辅以文献中论文的补充，本书可作为研究生算法分析导论性课程的基础来使用，或作为那些想要接近该领域文献的数学或计算机科学研究人员的参考书或自学基础材料。它还可以作为应用和方法的资源为字符组合数学和离散数学的学生和研究人员使用。

尽管算法数学分析方面的文献很多，但是关于普遍使用的方法和模型方面的基本信息尚不能很容易地为该领域的学生和研究人员直接使用。本书的目的就是处理这种情况，把想要提供给读者该领域问题的正确评价和学习正在发展以满足这些挑战的高级工具的需求背景的大量材料汇聚在一起。

背景要求。假设数学程度为在大学学习一年或两年的水平。组合数学和离散数学的基本课程可以提供有用的背景知识（可能与本书的某些内容重叠），实分析、数值方法或初等数论中的课程也是如此。我们将涉及所有这些领域，不过在这里只综述必需的材料，那些想要知道更多信息的读者可以参考标准的教科书。

假设编程经验为在大学学习过一个或两个学期的水平，包括初等数据结构。我们不在编程和算法实现问题上过多地耗费精力，算法和数据结构则是我们研究的中心课题。从某种意义上讲，我们的处理是完善的：对于基本的信息，我们给出概要的总结，必要时可以参考标准的教科书和一些基本的原始资料。

强烈推荐读者在数学计算时使用诸如MAPLE或Mathematica这样的计算机软件。当验证本书中的材料或求解习题结果时，这些软件系统可以帮助摆脱烦琐的计算。

相关的书籍。相关的教科书包括Knuth的*The Art of Computer Programming*；Gonnet和Baeza-Yates的*Handbook of Algorithms and Data Structure*；Sedgewick的*Algorithms*；Graham、Knuth和Patashnik的*Concrete Mathematics*；Cormen、Leiserson和Rivest的*Introduction to Algorithms*。本书可以看作是对这些著作的增补，下面依次评述。

实质上，本书最接近于Knuth的开创性著作。不过，我们的重点是在分析的数学方法上，而Knuth著作的范围广而全，算法的性质起着主要的作用，而分析的方法则是第二位的。本书可以作为Knuth的著作中所论及的先进结果的基础准备。

我们还讨论了自从Knuth的书出版以来始终在发展的算法分析中的方法和结果。Gonnet和Baeza-Yates的著作是这些结果的透彻综述，包括全面的参考文献；它主要介绍源自文献的一些相关结果。本书提供了通向该文献的基本预备内容。

我们也努力把重点放在基础、重要和有趣的算法上，像Sedgewick的*Algorithms*中所描述的一些算法，而Graham、Knuth和Patashnik的*Concrete Mathematics*则几乎全部侧重在数学方法方面。我们想要使本书成为Graham、Knuth和Patashnik的书中所讨论的基本数学方法和Sedgewick的书所涉及的基本算法的桥梁。

Cormen、Leiserson和Rivest的*Introduction to Algorithms*是那些提供通向算法的“设计和分析”研究文献著作的代表，一般是基于方法性能的粗略的最坏情形估计的。当需要更精确的结果时（对于大多数重要的方法），更复杂的模型和数学工具是必不可少的。Cormen、Leiserson和Rivest把重点放在算法的设计（通常带有为最坏情形性能定界的目的）上，而解析结果则用于帮助指导算法的设计。另一方面，我们的书是对他们的书以及类似的书的补充，因为我们侧重于算法的分析，特别是对那些能够用于获取可以预报算法性能的详细结果的方法。在这个过程中，我们也考虑与各种经典数学工具的关系，第1章全部用来描述这种环境。

本书还为其姐妹篇*Analytic Combinatorics*奠定了基础，那本书以更广的视野对本书涉及的材料进行一般的处理，并开发了高级方法和模型，这些方法和模型不仅在算法平均情形的分析中，而且在组合数学中，均可作为新的研究基础。对于阅读那本姐妹篇，需要更高水平的数学基础，相当于高年级本科生或低年级研究生所应具备的水平。当然，认真学习本书也足以具备这些基础。毫无疑问，使得本书充分有趣始终是我们的目的，读者将会惊喜地发现本书涉及更先进的材料。

如何使用本书。本书的读者在离散数学和计算机科学方面的基础很可能有相当大的差异。因此，了解书中的基本结构很有必要：全书总共八章，先是导论性的一章，接着的三章侧重于数学方法，最后四章集中讨论算法分析的应用：

引论

1. 算法分析概述

离散数学方法

2. 递归关系

3. 生成函数

4. 渐近逼近

算法与组合结构

5. 树

6. 排列

7. 串与trie树

8. 字与映射

第1章将书中材料恰当地进行概括，并帮助所有读者理解本书的基本目标以及其余各章在满足这些目标方面所担当的角色。第2章至第4章更侧重于数学，涉及离散数学的一些方法，

主要讨论一些基本概念和方法。第5章到第8章侧重于计算机科学，包含组合结构的一些性质，与基本算法的关系，以及若干解析结果。

虽然本书是自完备的，但是根据教师经验和学生的基础，可以有所侧重。一种方法是更偏重于数学，重点讲解本书前半部分的定理和证明，并介绍第5章到第8章的应用。另一种方法是更偏重于计算机科学，扼要地讨论第2章到第4章中的主要数学工具，而着重讨论本书第二部分的算法题材。不过，我们主要的意图是大部分的学生通过细心地通读本书，能够同时学到数学和计算机科学两个领域中新的东西。

具有较强计算机科学基础的学生可能已经见过本书后半部分的许多算法和数据结构，但在开始的时候他们的数学知识并不多；具有较强数学背景的学生很可能发现数学材料很熟悉，但算法和数据结构或许有些生疏。包括本书所有内容的课程可以帮助这两类学生根据他们已有的知识来填补知识缺口。

在每章末尾列出的参考文献以及散布在文中的数百道习题是对教材的补充，它们将鼓励读者更深刻地考虑书中的内容并考查原始的资料。

我们教授本书的经验已经证明，教师能够经常通过基于计算的实验课和家庭作业来补充讲义和阅读材料。这里所涵盖的内容对于学生使用诸如Mathematica或MAPLE这样的符号处理系统来深入研习专门知识是一个理想的构架。通过与经验研究比较而使数学研究得以确认的经验对于许多学生而言可能是非常有价值的。

致谢。我们非常感谢INRIA、普林斯顿大学和国家科学基金会，他们对本书提供了主要的支持。其他的支持由布朗大学、European Community (Alcom Project)、Institute for Defence Analyses、Ministère de la Recherche et de la Technologie、斯坦福大学、Université Libre de Bruxelles以及Xerox Palo Alto Research Center提供。本书耗费多年时间撰著，因此提供支持的个人和组织的名单很长，在此不能一一列出，深为抱歉。

Don Knuth对我们工作的影响极为重要，这从教材中明显可以看出。

普林斯顿、巴黎以及普罗维登斯大学的学生们这些年来对利用这些材料的早期版本授课的课程回馈了有益的信息。我们还要感谢Philippe Dumas、Mordecai Golin和他的学生San Kuen Chan、Ka Po Lam、Ngok Hing Leung、Derek Ka-Cheong Lueng和King Shan Lui，以及三位不具名评论家对手稿的详细评论。

R. S.

Ph. F.

1995年9月，Corfu

记号解释

- $\lfloor x \rfloor$ 下取整函数
小于或等于 x 的最大整数
- $\lceil x \rceil$ 上取整函数
大于或等于 x 的最小整数
- $\{x\}$ 小数部分
 $x - \lfloor x \rfloor$
- $\lg N$ 以2为底的对数
 $\log_2 N$
- $\ln N$ 自然对数
 $\log_e N$
- $\binom{n}{k}$ 二项式系数
从 n 项中选择 k 项的方法数
- $\left[\begin{matrix} n \\ k \end{matrix} \right]$ 第一类Stirling数
 n 个元素的具有 k 个圈的排列数
- $\left\{ \begin{matrix} n \\ k \end{matrix} \right\}$ 第二类Stirling数
将 n 个元素划分成 k 个非空子集的方法数
- ϕ 黄金分割
 $(1 + \sqrt{5})/2 = 1.618\ 03\dots$
- γ 欧拉常数
- σ Stirling函数
 $\sqrt{2\pi} = 2.506\ 62\dots$

目 录

出版者的话	
专家指导委员会	
译者序	
序	
前言	
记号解释	
第1章 算法分析概述	1
1.1 为什么要对算法进行分析	1
1.2 计算复杂性	2
1.3 算法分析的过程	6
1.4 平均情形分析	7
1.5 例:快速排序的分析	8
1.6 渐近逼近	13
1.7 分布	14
1.8 概率算法	16
参考文献	18
第2章 递归关系	21
2.1 基本性质	22
2.2 一阶递归	24
2.3 非线性一阶递归	26
2.4 高阶递归	28
2.5 求解递归的方法	32
2.6 二分分治递归和二进制数	37
2.7 一般的分治递归	43
参考文献	48
第3章 生成函数	51
3.1 常规生成函数	51
3.2 指数生成函数	55
3.3 利用生成函数求解递归	57
3.4 生成函数的展开	63
3.5 利用生成函数进行变换	65
3.6 关于生成函数的函数方程	67
3.7 利用OGF求解三数中值Quicksort递归	69
3.8 利用生成函数的计数	71
3.9 符号方法	74
3.10 拉格朗日反演	80
3.11 概率生成函数	82
3.12 二元生成函数	84
3.13 特殊函数	91
参考文献	94
第4章 渐近逼近	97
4.1 有关渐近逼近的记号	98
4.2 渐近展开式	102
4.3 渐近展开式的操作	107
4.4 有限和的渐近逼近	112
4.5 欧拉-麦克劳林求和	114
4.6 二元渐近性	119
4.7 拉普拉斯方法	128
4.8 算法分析中的“正态”例	131
4.9 算法分析中的“泊松”例	135
4.10 生成函数的渐近性	136
参考文献	140
第5章 树	141
5.1 二叉树	141
5.2 树和森林	143
5.3 树的性质	145
5.4 树的算法	147
5.5 二叉查找树	150
5.6 Catalan树中的平均路径长	153
5.7 二叉查找树中的路径长	156
5.8 随机树的可加参数	159
5.9 高	162
5.10 树性质平均情形结果的小结	167
5.11 树和二叉树的表示	168
5.12 无序树	172
5.13 标号树	178
5.14 其他类型的树	180
参考文献	186

第6章 排列	189	7.5 上下文无关语法	244
6.1 排列的基本性质	190	7.6 trie树	248
6.2 排列的算法	194	7.7 trie算法	251
6.3 排列的表示法	196	7.8 trie树的组合性质	254
6.4 计数问题	200	7.9 更大的字母表	258
6.5 利用CGF分析排列的性质	204	参考文献	259
6.6 逆序与插入排序	211	第8章 字与映射	263
6.7 左向右最小值与选择排序	216	8.1 使用分离链接的散列	263
6.8 圈与原位排列	220	8.2 字的基本性质	265
6.9 极值参数	223	8.3 生日悖论与赠券收藏家问题	269
参考文献	226	8.4 占有约束与极值参数	274
第7章 串与trie树	229	8.5 占有分布	278
7.1 串查找	230	8.6 开放定址散列法	284
7.2 位串的组合性质	232	8.7 映射	289
7.3 规则表达式	239	8.8 整数因子分解与映射	297
7.4 有限状态自动机与Knuth-Morris-Pratt算法	242	参考文献	301
		索引	303

第1章 算法分析概述

计算机算法性质的数学研究已经扩展到一个颇为广大的范畴，从一般复杂性研究直到特定解析结果的获得。在这一章，我们将讨论研究算法的各种手段中具有代表性的例子。同时，这些例子使我们得以介绍来自基本的和具有代表性领域即排序算法研究的各种经典的结果。

首先，我们将要考虑算法分析的一般动机，以及算法性能特征的各种研究方法之间的关系。其次，讨论计算复杂性，并把一个“最优的”排序算法，即归并排序（Mergesort）作为例子来考虑。之后，再对具有重要使用价值的排序算法的全面分析的主要方面进行考查，这个算法就是快速排序（Quicksort）。它包括研究对基本快速排序算法的各种改进，同时通过一些例子阐明，我们的分析是如何帮助人们调整参数以改进算法性能的。

本章将把我们的研究领域纳入到一些相关领域之中，并为本书其余部分的研究建立一个平台。在第2章到第4章，我们计划介绍基本的数学概念，这些概念是进行基本算法分析所必需的。第5章到第8章则讨论基本算法和数据结构的基本组合学性质。由于在计算机科学中使用的基本方法和经典数学分析之间存在着紧密的联系，因此我们在本书中同时讨论来自这两个方面的某些介绍性的材料。

1.1 为什么要对算法进行分析

关于这个基本的问题，有几种不同的答案，它们依赖于算法的环境：算法的预期使用；从实用性和理论的立场上看，一个算法与其他算法相比的价值；分析的困难程度和所要求的答案的精确性。

分析算法的最直接的原因是找出它的特性，以便估计它对各种应用的适用程度，或在相同的应用中与其他算法进行比较。我们注意的特征最常见的是时间和空间这样一些基本的资源，尤其是时间。简单地说，我们想要知道在一台特定的计算机上实现一个特定算法要运行多长时间，以及需要多少空间。一般说来，分析是相对独立于特定实现方法的——我们侧重于得到算法基本性能的相关结果，这些算法能够用来算出在具体的机器上对于实际资源需求的精确估计。

实践中，在算法和它的实现方法的特性之间达到完全独立做起来恐怕不容易。算法实现的质量、编译器的特性、机器的结构以及编程环境的其他主要方面，都将对性能产生剧烈的影响。我们必须认识到这些影响，这样才能保证分析的结果是有用的。另一方面，在某些情况下，算法分析可以帮助识别那些能够充分利用编程环境的方法。

偶尔有些性质很重要，但它们并不是时间和空间，因此我们分析的重点也要相应地改变。例如，驱动电路布线装置的算法可能用以确定所用线的总量，数值问题的算法可能要确定它提供的答案的准确程度。再有，有时候在分析中使用成倍的资源是可取的。例如，使用大量内存的算法可能比使用很少内存也能得到的算法所耗费的时间少得多。实际上，进行仔细分析的一个主要的动机就是提供准确的信息，以帮助在这样一些情况下作出正确的权衡决策。

术语算法分析（analysis of algorithms）一直用于描述两种相当不同的方法，这两种方法把计算机程序的研究置于科学的基础之上，现在我们对它们依次进行讨论。

第一种方法由Aho、Hopcroft和Ullman[2]所提倡，重点研究确定算法最坏情形性能的增长（“上界”）。这种分析的主要目标就是确定在下述意义下哪些算法是最优的：它们对同一问题的任何算法最坏情形的性能可以证明一个匹配的“下界”。有时称这种类型的分析为计算复杂性（computational complexity），不过，这个术语留给对问题、算法、语言和机器之间关系的一般研究或许更妥当。

处理算法分析的第二种方法由Knuth[13][14][15][17]提出，主要通过确定最佳情形、最坏情形和平均情形的性能来精确地刻画算法的性能，使用的是一套可以精化的方法，在需要的时候能够得出更加精确的答案。这种分析的主要目标是当运行在特定的计算机上的时候，能够准确地预测特定算法的性能特征。

我们可以把这两种方法看作是有效算法的设计和分析的必需阶段。当面对求解一个新问题的新算法的时候，我们的兴趣在于形成这样一种粗略的认识：新算法预期能有多好，对于同一问题它比其他的算法如何。计算复杂性研究能够提供这种认识。然而，在这样一种粗略的复杂性分析中丧失如此多的精度，以致提供不了使我们预测算法具体实现的性能的特定信息，也提供不了将一个算法与另外的算法进行正确比较所需要的特定信息。为了能够进行这些工作，我们需要关于算法实现、所使用的计算机，以及本书我们将要看到的，由算法使用的一些结构的数学性质的全部细节。计算复杂性可以看成是形成更精练、更准确分析的正在进行的过程的第一步；我们更愿意使用术语算法分析来表示整个的过程，其目标是提供所需任意精度的解决方案。

对算法的分析能够帮助我们对算法更深入地理解，并能够提出有根有据的改进。算法越复杂，分析越困难。在分析过程中，算法趋向于变得更短小、更简单和更讲究。尤为重要的是，正确分析所需要的仔细全面的审查常常使得算法更好和更有效地实现。分析要求对算法全面的理解，这种理解比只是简单得出行得通的实现方法远为全面和深刻。实际上，当解析的结果与经验研究一致的时候，我们将会确信算法的合理性以及分析过程的正确性。

有些算法是值得分析的，因为它们的分析能够增强现有的数学工具集。这样一些算法也许实用价值有限，但可能具有类似于实用价值的算法的性质，使得理解它们能够有助于理解未来更重要的方法。

另一方面，许多算法（有些有很强的实用价值，有些则很少甚至没有实用价值）具有复杂的性能结构，这种结构具有独立的数学意义。其算法分析产生的组合学问题的活跃成分导致一些具有挑战性的重要的数学问题，这些问题扩展了经典组合学的范畴，有助于揭示计算机程序的性质。

为了清楚地表达这些思想，下面首先从计算复杂性的角度，然后再以算法分析的观点，详细考虑某些经典的结果。作为阐述不同观点的实际例子，我们考查一些排序算法（sorting algorithms），它们以数值顺序、字母顺序、或其他顺序重新排列一组数据。这是一个重要的实际问题，该问题仍然是广泛研究的目标，因为排序程序在许多应用中起着核心的作用。

1.2 计算复杂性

计算复杂性的主要目标是把算法按照它们的性能特征进行分类。第一步是建立表达结果的相应数学概念：

定义 给定函数 $f(N)$ ，则

$O(f(N))$ 表示所有 $g(N)$ 的集合，其中 $g(N)$ 满足当 $N \rightarrow \infty$ 时， $|g(N)/f(N)|$ 上有界。

$\Omega(f(N))$ 表示所有 $g(N)$ 的集合, 其中 $g(N)$ 满足当 $N \rightarrow \infty$ 时, $|g(N)/f(N)|$ 以一个 (严格的) 正数为其下界。

$\Theta(f(N))$ 表示所有 $g(N)$ 的集合, 其中 $g(N)$ 满足当 $N \rightarrow \infty$ 时, $|g(N)/f(N)|$ 既上有界又下有界。

这些概念取自经典数论, 由 Knuth 在 1976 年的一篇论文 [16] 的算法分析中提倡使用。在对算法性能的界的数学表述方面, 这些记号得到了广泛的使用。

大 O 记法提供一种表达上界的方法; 大 Ω 记法提供一种表达下界的方法; 而大 Θ 记法则提供一种表达同时有上界又有下界的方法。大 O 记法在数学中最普通的用法是表示渐近近似性质, 我们将在第 4 章对它进行详细的考察。这个记号在算法分析中也比其余两个记号使用得普遍, 在描述算法运行时间的表达式中, 它一般表示相对小“错误”的项。记号 Ω 和 Θ 直接与计算复杂性相关, 不过类似的记号在其他的应用领域也在使用 (见 [16])。

对于复杂性研究, 这些记号的重要性在于, 它们实现的细节被忽略的常数因子所隐藏。由于常数因子被忽略, 因此使用这些记法的副作用在于, 使用它们得到的数学结果要比寻找更精确的答案简单。例如, “自然”对数 $\ln N \equiv \log_e N$ 和“二进制”对数 $\lg N \equiv \log_2 N$ 经常出现, 但是, 它们之间的关系是相差一个常数因子, 因此, 在复杂性分析中我们可以把二者都看成是 $O(\log N)$ 。

习题 1.1 证明 $f(N) = N \lg N + O(N)$ 意味着 $f(N) = \Theta(N \log N)$ 。

作为使用这些记号研究算法性能特征的一个例子, 我们考虑将数组 $a[1..N]$ 中的数集排序的方法。输入是数组中的数, 顺序是任意的并且是未知的; 输出为该数组中同样这些数, 但以递升的顺序排列。这是一个得到深入研究的基本问题: 我们将考虑求解该问题的一个算法, 然后证明, 从精确的技术意义上看, 算法是“最优的”。

首先, 我们要证明, 使用著名的递归算法, 即归并排序 (Mergesort) 能够有效地解决这个排序问题。归并排序以及本书中处理的几乎所有的算法均在 Sedgewick [19] 中有详细的讨论, 因此我们这里只给出简要的描述。对于各种算法、实现和应用的进一步细节有兴趣的读者, 可以参考 Cormen、Leiserson 和 Rivest [5], Gonnet 和 Baeza-Yates [8], Knuth [13][14][15], Sedgewick [19] 等人的著作以及其他一些资料。

归并排序将数组从中间分开, (递归地) 将两部分排序, 然后将所得到的已排序的子文件合并到一起从而得到排序结果。为此, 该算法使用了两个辅助数组 b 和 c , 如程序 1.1 所示。通过 `mergesort(1, N)` 调用该过程将数组 $a[1..N]$ 排序。在递归调用之后, 排序的前一半 $a[1..m]$ 被拷贝到辅助数组 $b[1..m-1+1]$, 而排序的后一半 $a[m+1..r]$ 被拷贝到第 2 个辅助数组 $c[1..r-m]$ 。然后, 通过把元素 $b[i]$ 和 $c[j]$ 中的较小者移到 $a[k]$, 相应地使 i 或 j 增 1 来完成合并。程序用到“警戒标记” `max` 以帮助完成当一个辅助数组已经没有元素时将另一个辅助数组剩下的部分移回到 a , 此处假设 `max` 比所有的元素都大。

习题 1.2 实现一种归并排序, 该排序将数组分成 3 个相等的部分, 将这 3 部分排序, 并进行 3-路合并。凭经验比较该算法与标准的归并排序的运行时间。

归并排序是著名的分治算法 (divide-and-conquer) 设计范例的原型, 其中, 一个问题的解决是通过 (递归地) 求解更小的一些子问题并利用这些子问题的解来求解原问题而得到的。在本书中, 我们将分析许多这样的算法。更一般地, 我们将看到象归并排序这样的算法的递归结构如何直接导致它们性能特征的数学描述。

在当前环境下, 归并排序是重要的方法, 因为它能够保证不亚于任何排序算法的效率。为了更精确地表达这个结论, 我们开始分析归并排序运行时间的控制因素, 即它所用到的比