

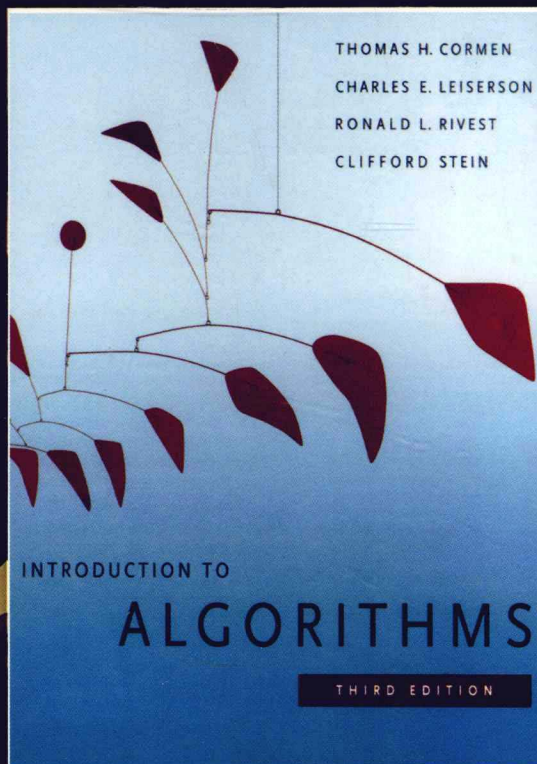
原书第3版

算法导论

(美) Thomas H. Cormen Charles E. Leiserson 著
Ronald L. Rivest Clifford Stein

殷建平 徐云 王刚 刘晓光 苏明 邹恒明 王宏志 译

Introduction to Algorithms
Third Edition



机械工业出版社
China Machine Press

计 算 丛 书

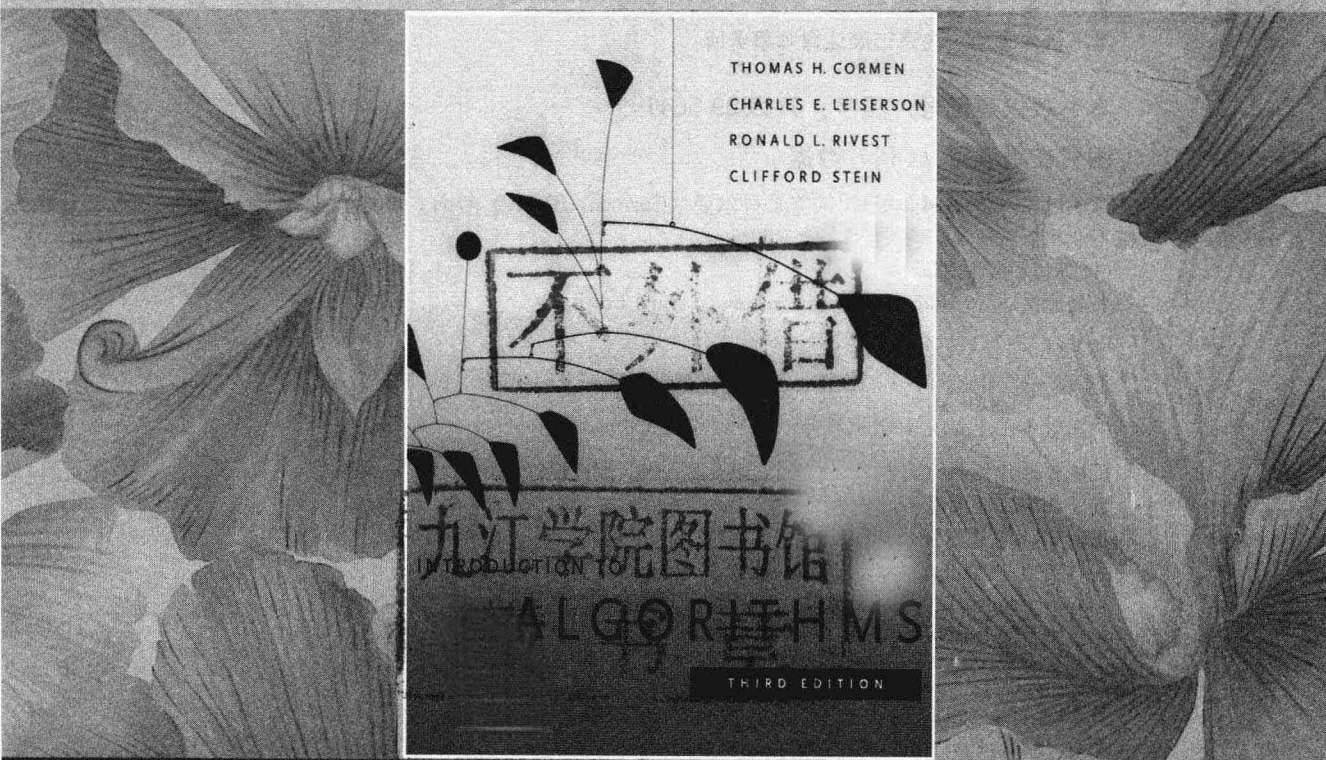
原书第3版

算法导论

(美) Thomas H. Cormen Charles E. Leiserson 著
Ronald L. Rivest Clifford Stein

殷建平 徐云王 刚 刘晓光 苏明 邹恒明 王宏志 译

Introduction to Algorithms
Third Edition



机械工业出版社
China Machine Press

本书提供了对当代计算机算法研究的一个全面、综合性的介绍。全书共八部分，内容涵盖基础知识、排序和顺序统计量、数据结构、高级设计和分析技术、高级数据结构、图算法、算法问题选编，以及数学基础知识。书中深入浅出地介绍了大量的算法及相关的知识结构，以及用于解决一些复杂计算问题的高级策略（如动态规划、贪心算法、摊还分析等），重点在于算法的分析与设计。对于每一个专题，作者都试图提供目前最新的研究成果及样例解答，并通过清晰的图示来说明算法的执行过程。此外，全书包含 957 道练习和 158 道思考题，并且作者在网站上给出了部分题的答案。

本书内容丰富，叙述深入浅出，适合作为计算机及相关专业本科生数据结构课程和研究生算法课程的教材，同时也适合专业技术人员参考使用。

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein: Introduction to Algorithms, Third Edition (ISBN 978-0-262-03384-8). Original English language edition copyright © 2009 by Massachusetts Institute of Technology. Simplified Chinese Translation Copyright © 2013 by China Machine Press.

Simplified Chinese translation rights arranged with MIT Press through Bardonn-Chinese Media Agency.

No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or any information storage and retrieval system, without permission, in writing, from the publisher. All rights reserved.

本书中文简体字版由 MIT Press 通过 Bardonn-Chinese Media Agency 授权机械工业出版社在中华人民共和国境内独家出版发行。未经出版者书面许可，不得以任何方式抄袭、复制或节录本书中的任何部分。

封底无防伪标均为盗版
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：01-2009-5641

图书在版编目（CIP）数据

算法导论（原书第 3 版）/（美）科尔曼（Cormen, T. H.）等著；殷建平等译. —北京：机械工业出版社，2013.1

（计算机科学丛书）

书名原文：Introduction to Algorithms, Third Edition

ISBN 978-7-111-40701-0

I. 算… II. ①科… ②殷… III. 电子计算机—算法理论 IV. TP301.6

中国版本图书馆 CIP 数据核字（2012）第 290499 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：王春华

北京京师印务有限公司印刷

2013 年 1 月第 1 版第 1 次印刷

185mm×260mm·49.75 印张

标准书号：ISBN 978-7-111-40701-0

定价：128.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：(010) 88378991；88361066

购书热线：(010) 68326294；88379649；68995259

投稿热线：(010) 88379604

读者信箱：hzjsj@hzbook.com

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，还揭示了学术的源变，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和发展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到积极的推动作用，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公司较早意识到“出版要为教育服务”。自1998年开始，我们就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力，我们与 Pearson, McGraw-Hill, Elsevier, MIT, John Wiley & Sons, Cengage 等世界著名出版公司建立了良好的合作关系，从他们现有的数百种教材中甄选出 Andrew S. Tanenbaum, Bjarne Stroustrup, Brain W. Kernighan, Dennis Ritchie, Jim Gray, Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, Abraham Silberschatz, William Stallings, Donald E. Knuth, John L. Hennessy, Larry L. Peterson 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、研究及珍藏。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相助，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。其影印版“经典原版书库”作为姊妹篇也被越来越多实施双语教学的学校所采用。

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。随着计算机科学与技术专业学科建设的不断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将步入一个新的阶段，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。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，我们的联系方式如下：

华章网站：www.hzbook.com

电子邮件：hzsj@hzbook.com

联系电话：(010) 88379604

联系地址：北京市西城区百万庄南街1号

邮政编码：100037



华章教育

华章科技图书出版中心

我从1994年开始每年都为本科生讲授“算法设计与分析”课程，粗略地统计一下，发现至今已有5000余名各类学生听过该课。算法的重要性不言而喻，因为不管新概念、新方法、新理论如何引人注目，信息的表示与处理总是计算技术（含软件、硬件、应用、网络、安全、智能等）永恒的主题。信息处理的核心是算法。在大数据时代，设计高效的算法显得格外重要。

当初，为了教好这门基础必修课，提高教学质量，我觉得应该从教学内容的改革入手，具体来说，采用的教材应该与国际一流大学接轨。1997年访美期间，在Stanford大学了解到他们采用的教材是Thomas H. Cormen等人著的《Introduction to Algorithms》，于是从Stanford书店买了一本带回来，从第二年便开始改用该书作为教材。至今，15年过去了，我们一直追随其变迁，从第2版到第3版。教学实践证明它确实是一本好教材，难怪世界范围内包括MIT、CMU、Stanford、UCB、Cornell、UIUC等国际、国内名校在内的1000余所大学都一直用它作为教材或教学参考书，也难怪它印数巨大且在《高引用计算机科学文献》（《Most Cited Computer Science Citations》）一览表中名列前茅。

这本书的原有1200多页，内容非常丰富，不但涵盖了典型算法、算法分析、算法设计方法和NP完全等内容，而且还包括数据结构，甚至高级数据结构的介绍。后者可以作为国内“数据结构”课程的教材或教学参考资料。在学时有限的情况下，要在本科阶段教完前者的所有内容也是困难的，故要做取舍。好在该书的各个章节相对独立且难度由浅入深，我们的做法是将相对容易的、一般的入门性内容留在本科阶段，而将相对难的、专门的、较深入的内容并入研究生课程“算法及复杂性”或“计算复杂性”。除本校外，本人就曾多次应邀在兰州大学、湖南大学和浙江师范大学等院校为研究生讲授过这些内容。其实该书也适合希望增强自身程序设计能力和程序评判能力的广大应用计算技术的社会公众，特别是参加信息学奥林匹克竞赛和ACM程序设计竞赛的选手及其教练员。

在教学过程中，我们发现该书具有以下特点：1) 选材与时俱进，具有实用性且能引起读者的兴趣。该书中研究的许多问题都是当前现实应用中的关键技术问题。2) 采用伪代码描述算法，既简洁易懂又便于抓住本质，再配上丰富的插图来描述和解释算法的执行过程，使得教学内容更加通俗，便于自学。3) 对算法正确性和复杂度的分析比较全面，既有严密的论证，又有直观的解释。4) 既有结论性知识的介绍，也有逐步导出结论的研究过程的展示。5) 丰富的练习题和思考题使得及时检验所学知识掌握情况和进一步拓展学习内容成为可能。

在第3版的《Introduction to Algorithms》出版后，我们应机械工业出版社编辑的邀请，启动了长久的翻译工程，先后参加翻译工作的老师有：国防科学技术大学的殷建平教授（翻译第1~3章）、中国科学技术大学的徐云教授（翻译第10~14章、第18~21章和第27章）、南开大学的王刚教授（翻译第4章和第15~17章）、南开大学的刘晓光教授（翻译第6~9章）、南开大学的苏明副研究员（翻译第5章和第28~30章）、上海交通大学的邹恒明教授（翻译第22~26章）、哈尔滨工业大学的王宏志副教授（翻译第31~35章和附录部分）。由于水平有限且工作量巨大，译文中一定存在许多不足，在此敬请各位同行专家学者和广大读者批评指正，欢迎大家将发现的错误或提出的意见与建议发送到邮箱：algorithms@hzbook.com。在整个工程即将完成之际，我们要特别感谢潘金贵、顾铁成、李成法和叶懋等参与本书第2版翻译的老师，是他们使得这本重要教材在国内有了广泛读者。同时也要感谢机械工业出版社的温莉芳编辑和王春华编辑，没有你们的信任、耐心和支持，整个翻译工作不可能完成。

殷建平

2012年11月于长沙

在计算机出现之前，就有了算法。现在有了计算机，就需要更多的算法，算法是计算的核心。

本书提供了对当代计算机算法研究的一个全面、综合的介绍。书中给出了多个算法，并对它们进行了较为深入的分析，使得这些算法的设计和分析易于被各个层次的读者所理解。我们力求在不牺牲分析的深度和数学严密性的前提下，给出深入浅出的说明。

书中每一章都给出了一个算法、一种算法设计技术、一个应用领域或一个相关的主题。算法是用英语和一种“伪代码”来描述的，任何有一点程序设计经验的人都能看得懂。书中给出了 244 幅图，说明各个算法的工作过程。我们强调将算法的效率作为一种设计标准，对书中的所有算法，都给出了关于其运行时间的详细分析。

本书主要供本科生和研究生的算法或数据结构课程使用。因为书中讨论了算法设计中的工程问题及其数学性质，所以，本书也可以供专业技术人员自学之用。

本书是第 3 版。在这个版本里，我们对全书进行了更新，包括新增了若干章、修订了伪代码等。

致使用本书的教师

本书的设计目标是全面、适用于多种用途。它可用于若干课程，从本科生的数据结构课程到研究生的算法课程。由于书中给出的内容比较多，只讲一学期一般讲不完，因此，教师们应该将本书看成是一种“缓存区”或“瑞典式自助餐”，从中挑选出能最好地支持自己希望教授的课程的内容。

教师们会发现，要围绕自己所需的各个章节来组织课程是比较容易的。书中的各章都是相对独立的，因此，你不必担心意想不到的或不必要的各章之间的依赖关系。每一章都是以节为单位，内容由易到难。如果将本书用于本科生的课程，可以选用每一章的前面几节内容；用于研究生的课程中，则可以完整地讲授每一章。

全书包含 957 道练习和 158 道思考题。每一节结束时给出练习，每一章结束时给出思考题。练习一般比较短，用于检查学生对书中内容的基本掌握情况。有一些是简单的自查性练习，有一些则要更充实，可以作为家庭作业布置给学生。每一章后的思考题都是一些叙述较为详细的实例研究，它们常常会介绍一些新的知识。一般来说，这些思考题都会包含几个小问题，引导学生逐步得到问题的解。

根据本书前几版的读者反馈，我们在本书配套网站上公布了其中一些练习和思考题的答案（但不是全部），网址为 <http://mitpress.mit.edu/algorithms/>。我们会定期更新这些答案，因此需要教师每次授课前都到这个网站上来查看。

在那些不太适合本科生、更适合研究生的章节和练习前面，都加上了星号（*）。带星号的章节也不一定就比不带星号的更难，但可能要求了解更多的数学知识。类似地，带星号的练习可能要求有更好的数学背景或创造力。

致使用本书的学生

希望本教材能为学生提供关于算法这一领域的有趣介绍。我们力求使书中给出的每一个算法都易于理解和有趣。为了在学生遇到不熟悉或比较困难的算法时提供帮助，我们逐个步骤地描述每一个算法。此外，为了便于大家理解书中对算法的分析，对于其中所需的数学知识，我们

给出了详细的解释。如果对某一主题已经有所了解，会发现根据书中各章的编排顺序，可以跳过一些介绍性的小节，直接阅读更高级的内容。

本书是一本大部头著作，学生所修的课程可能只讲授其中的一部分。我们试图使它能成为一本现在对学生有用的教材，并在其将来的职业生涯中，也能成为一本案头的数学参考书或工程实践手册。

阅读本书需要哪些预备知识呢？

- 需要有一些程序设计方面的经验，尤其需要理解递归过程和简单的数据结构，如数组和链表。
- 应该能较为熟练地利用数学归纳法进行证明。书中有一些内容要求学生具备初等微积分方面的知识。除此之外，本书的第一部分和第八部分将介绍需要用到所有数学技巧。

我们收到学生的反馈，他们强烈希望提供练习和思考题的答案，为此，我们在 <http://mitpress.mit.edu/algorithms/> 这个网站上给出了少数练习和思考题的答案，学生可以根据我们的答案来检验自己的解答。

致使用本书的专业技术人员

本书涉及的主题非常广泛，因而是一本很好的算法参考手册。因为每一章都是相对独立的，所以读者可以重点查阅自己感兴趣的专题。

在我们所讨论的算法中，多数都有着极大的实用价值。因此，我们在书中涉及了算法实现方面的考虑和其他工程方面的问题。对于那些为数不多的、主要具有理论研究价值的算法，通常还给出其实用的替代算法。

如果希望实现这些算法中的任何一个，你会发现将书中的伪代码翻译成你熟悉的某种程序设计语言是一件相当直接的事。伪代码被设计成能够清晰、简明地描述每一个算法。因此，我们不考虑错误处理和其他需要对读者所用编程环境有特定假设的软件工程问题。我们力求简单而直接地给出每一个算法，而不会让某种特定程序设计语言的特殊性掩盖算法的本质内容。

如果你是在课堂外使用本书，那么可能无法从教师那里得到答案来验证自己的解答，因此，我们在 <http://mitpress.mit.edu/algorithms/> 这个网站上给出了部分练习和思考题的答案，读者可以免费下载参考。

致我们的同事

我们在本书中给出了详尽的参考文献。每一章在结束时都给出了“本章注记”，介绍一些历史性的细节和参考文献。但是，各章的注记并没有提供整个算法领域的全部参考文献。有一点可能是让人难以置信的，即使是在本书这样一本大部头中，由于篇幅的原因，很多有趣的算法都没能包括进来。

尽管学生们发来了大量的请求，希望我们提供思考题和练习的解答，但我们还是决定基本上不提供思考题和练习的参考答案（少数除外），以打消学生们试图查阅答案，而不是自己动手得出答案的念头。

第3版中所做的修改

在本书的第2版和第3版之间有哪些变化呢？这两版之间的变化量和第2版与第1版之间的变化量相当，正如在第2版的前言中所说，这些版本之间的变化可以说不太大，也可以说很大，具体要看读者怎么看待这些变化了。

快速地浏览一遍目录，你就会发现，第2版中的多数章节在第3版中都出现了。在第3版

中，去掉了两章和一节的内容，新增了三章以及两节的内容。如果单从目录来判断第3版中改动的范围，得出的结论很可能是改动不大。

我们依然保持前两版的组织结构，既按照问题领域又根据技术来组织章节内容。书中既包含基于技术的章，如分治法、动态规划、贪心算法、摊还分析、NP完全性和近似算法，也包含关于排序、动态集的数据结构和图问题算法的完整部分。我们发现虽然读者需要了解如何应用这些技术来设计和分析算法，但是思考题中很少提示应用哪个技术来解决这些问题。

下面总结了第3版的主要变化：

- 新增了讨论 van Emde Boas 树和多线程算法的章节，并且将矩阵基础移至附录。
- 修订了递归式那一章的内容，更广泛地覆盖分治法，并且前两节介绍了应用分治法解决两个问题。4.2节介绍了用于矩阵乘法的 Strassen 算法，关于矩阵运算的内容已从本章移除。
- 移除两章很少讲授的内容：二项堆和排序网络。排序网络中的关键思想——0-1原理，在本版的思考题 8-7 中作为比较交换算法的 0-1 排序引理进行介绍。斐波那契堆的处理不再依赖二项堆。
- 修订了动态规划和贪心算法相关内容。与第2版中的装配线调度问题相比，本版用一个更有趣的问题——钢条切割来引入动态规划。而且，我们比在第2版中更强调助记性，并且引入子问题图这一概念来阐释动态规划算法的运行时间。在我们给出的贪心算法例子（活动选择问题）中，我们以更直接的方式给出贪心算法。
- 我们从二叉搜索树（包括红黑树）删除一个结点的方式，现在保证实际所删除的结点就是请求删除的结点（在前两版中，有些情况下某个其他结点可能被删除）。用这种新的方式删除结点，如果程序的其他部分保持指针指向树中的结点，那么终止时就不会错误地将指针指向已删去的结点。
- 流网络相关材料现在基于边上的全部流。这种方法比前两版中使用的净流更直观。
- 由于关于矩阵基础和 Strassen 算法的材料移到了其他章，矩阵运算这一章的内容比第2版中所占的篇幅更小。
- 修改了对 Knuth-Morris-Pratt 字符串匹配算法的讨论。
- 修正了上一版中的一些错误。在网站上，这些错误大多数都已在第2版的勘误中给出，但是有些没有给出。
- 根据许多读者的要求，我们改变了书中伪代码的语法，现在用“=”表示赋值，用“==”表示检验相等，正如 C、C++、Java 和 Python 所用的。同样，我们不再使用关键字 **do** 和 **then** 而是使用“//”作为程序行末尾的注释符号。我们现在还使用点标记法表明对象属性。书中的伪代码仍是过程化的，而不是面向对象的。换句话说，我们只是简单地调用过程，将对象作为参数传递，而不是关于对象的运行方法。
- 新增 100 道练习和 28 道思考题，还更新并补充了参考文献。
- 最后，我们对书中的语句、段落和小节进行了一些调整，以使本书条理更清晰。

网站

读者可以通过 <http://mitpress.mit.edu/algorithms/> 这个网站来获取补充资料，以及与我们联系。这个网站上给出了已知错误的清单、部分练习和思考题的答案等。此外，网站上还告诉读者如何报告错误或者提出建议。

第3版致谢

我们已经与 MIT Press 合作 20 多年，建立了很好的合作关系！感谢 Ellen Faran、Bob Prior、

Ada Brunstein 和 Mary Reilly 的帮助和支持。

在出版第 3 版时，我们在达特茅斯学院计算机科学系、MIT 计算机科学与人工智能实验室、哥伦比亚大学工业工程与运筹学系从事教学和科研工作。感谢这些学校和同事为我们提供的支持和实验环境。

Julie Sussman, P. P. A 担当本书第 3 版的技术编辑，再次拯救了我们。每次审阅，我们都觉得已经消除了错误，但是 Julie 还是发现了许多错误。她还帮我们改进了几处文字表述。如果有技术编辑名人堂，Julie 一定第一轮就可以入选。Julie 是非凡的，我们怎么感谢都是不够的。Priya Natarajan 也发现了一些错误，使得我们可以在将本书交给出版社前修正这些错误。书中的任何错误（毫无疑问，一定存在一些错误）都由作者负责（或许这些错误有些是 Julie 审阅材料后引入的）。

对于 van Emde Boas 树的处理出自于 Erik Demaine 的笔记，转而也受到 Michael Bender 的影响。此外，我还将 Javed Aslam、Bradley Kuszmaul 和 Hui Zha 的思想也整合到这一版。

多线程算法这一章是基于与 Harald Prokop 一起撰写的笔记，其他在 MIT 从事 Cilk 项目的同事也对本部分内容有所贡献，包括 Bradley Kuszmaul 和 Matteo Frigo。多线程伪代码的设计灵感来自 MIT Cilk 扩展到 C，以及由 Cilk Arts 的 Cilk++ 扩展到 C++。

我们还要感谢许多第 1 版和第 2 版的读者，他们报告了所发现的错误，或者提出了改进本书的建议。我们修正了全部报告来的真实错误，并且尽可能多地采纳了读者的建议。我们很高兴有这么多人 为本书做出贡献，但是很遗憾我们无法全部列出这些贡献者。

最后，非常感谢我们各自的妻子 Nicole Cormen、Wendy Leiserson、Gail Rivest 和 Rebecca Ivry，还有我们的孩子 Ricky、Will、Debby、Katie Leiserson、Alex、Christopher Rivest，以及 Molly、Noah 和 Benjamin Stein。感谢他们在我们写作本书过程中给予的爱和支持。正是由于有了来自家庭的耐心和鼓励，本书的写作工作才得以完成。谨将此书献给他们。

Thomas H. Cormen, 新罕布什尔州黎巴嫩市

Charles E. Leiserson, 马萨诸塞州剑桥市

Ronald L. Rivest, 马萨诸塞州剑桥市

Clifford Stein, 纽约州纽约市

出版者的话

译者序

前言

第一部分 基础知识

第 1 章 算法在计算中的作用	3
1.1 算法	3
1.2 作为一种技术的算法	6
思考题	8
本章注记	8
第 2 章 算法基础	9
2.1 插入排序	9
2.2 分析算法	13
2.3 设计算法	16
2.3.1 分治法	16
2.3.2 分析分治算法	20
思考题	22
本章注记	24
第 3 章 函数的增长	25
3.1 渐近记号	25
3.2 标准记号与常用函数	30
思考题	35
本章注记	36
第 4 章 分治策略	37
4.1 最大子数组问题	38
4.2 矩阵乘法的 Strassen 算法	43
4.3 用代入法求解递归式	47
4.4 用递归树方法求解递归式	50
4.5 用主方法求解递归式	53
* 4.6 证明主定理	55
4.6.1 对 b 的幂证明主定理	56
4.6.2 向下取整和向上取整	58
思考题	60
本章注记	62
第 5 章 概率分析和随机算法	65
5.1 雇用问题	65

5.2 指示器随机变量	67
5.3 随机算法	69
* 5.4 概率分析和指示器随机变量的 进一步使用	73
5.4.1 生日悖论	73
5.4.2 球与箱子	75
5.4.3 特征序列	76
5.4.4 在线雇用问题	78
思考题	79
本章注记	80

第二部分 排序和顺序统计量

第 6 章 堆排序	84
6.1 堆	84
6.2 维护堆的性质	85
6.3 建堆	87
6.4 堆排序算法	89
6.5 优先队列	90
思考题	93
本章注记	94
第 7 章 快速排序	95
7.1 快速排序的描述	95
7.2 快速排序的性能	97
7.3 快速排序的随机化版本	100
7.4 快速排序分析	101
7.4.1 最坏情况分析	101
7.4.2 期望运行时间	101
思考题	103
本章注记	106
第 8 章 线性时间排序	107
8.1 排序算法的下界	107
8.2 计数排序	108
8.3 基数排序	110
8.4 桶排序	112
思考题	114
本章注记	118

第 9 章 中位数和顺序统计量	119	思考题	187
9.1 最小值和最大值	119	本章笔记	191
9.2 期望为线性时间的选择算法	120	第 14 章 数据结构的扩张	193
9.3 最坏情况为线性时间的选择 算法	123	14.1 动态顺序统计	193
思考题	125	14.2 如何扩张数据结构	196
本章笔记	126	14.3 区间树	198
第三部分 数据结构		思考题	202
第 10 章 基本数据结构	129	本章笔记	202
10.1 栈和队列	129	第四部分 高级设计和分析技术	
10.2 链表	131	第 15 章 动态规划	204
10.3 指针和对象的实现	134	15.1 钢条切割	204
10.4 有根树的表示	137	15.2 矩阵链乘法	210
思考题	139	15.3 动态规划原理	215
本章笔记	141	15.4 最长公共子序列	222
第 11 章 散列表	142	15.5 最优二叉搜索树	226
11.1 直接寻址表	142	思考题	231
11.2 散列表	143	本章笔记	236
11.3 散列函数	147	第 16 章 贪心算法	237
11.3.1 除法散列法	147	16.1 活动选择问题	237
11.3.2 乘法散列法	148	16.2 贪心算法原理	242
* 11.3.3 全域散列法	148	16.3 赫夫曼编码	245
11.4 开放寻址法	151	* 16.4 拟阵和贪心算法	250
* 11.5 完全散列	156	* 16.5 用拟阵求解任务调度问题	253
思考题	158	思考题	255
本章笔记	160	本章笔记	257
第 12 章 二叉搜索树	161	第 17 章 摊还分析	258
12.1 什么是二叉搜索树	161	17.1 聚合分析	258
12.2 查询二叉搜索树	163	17.2 核算法	261
12.3 插入和删除	165	17.3 势能法	262
12.4 随机构建二叉搜索树	169	17.4 动态表	264
思考题	171	17.4.1 表扩张	265
本章笔记	173	17.4.2 表扩张和收缩	267
第 13 章 红黑树	174	思考题	270
13.1 红黑树的性质	174	本章笔记	273
13.2 旋转	176	第五部分 高级数据结构	
13.3 插入	178	第 18 章 B 树	277
13.4 删除	183	18.1 B 树的定义	279

18.2 B 树上的基本操作	281	22.5 强连通分量	357
18.3 从 B 树中删除关键字	286	思考题	360
思考题	288	本章笔记	361
本章笔记	289	第 23 章 最小生成树	362
第 19 章 斐波那契堆	290	23.1 最小生成树的形成	362
19.1 斐波那契堆结构	291	23.2 Kruskal 算法和 Prim 算法	366
19.2 可合并堆操作	292	思考题	370
19.3 关键字减值和删除一个结点	298	本章笔记	373
19.4 最大度数的界	300	第 24 章 单源最短路径	374
思考题	302	24.1 Bellman-Ford 算法	379
本章笔记	305	24.2 有向无环图中的单源最短路径 问题	381
第 20 章 van Emde Boas 树	306	24.3 Dijkstra 算法	383
20.1 基本方法	306	24.4 差分约束和最短路径	387
20.2 递归结构	308	24.5 最短路径性质的证明	391
20.2.1 原型 van Emde Boas 结构	310	思考题	395
20.2.2 原型 van Emde Boas 结构上的操作	311	本章笔记	398
20.3 van Emde Boas 树及其操作	314	第 25 章 所有结点对的最短路径 问题	399
20.3.1 van Emde Boas 树	315	25.1 最短路径和矩阵乘法	400
20.3.2 van Emde Boas 树的操作	317	25.2 Floyd-Warshall 算法	404
思考题	322	25.3 用于稀疏图的 Johnson 算法	409
本章笔记	323	思考题	412
第 21 章 用于不相交集的数据 结构	324	本章笔记	412
21.1 不相交集的操作	324	第 26 章 最大流	414
21.2 不相交集的链表表示	326	26.1 流网络	414
21.3 不相交集森林	328	26.2 Ford-Fulkerson 方法	418
* 21.4 带路径压缩的按秩合并的 分析	331	26.3 最大二分匹配	428
思考题	336	* 26.4 推送-重贴标签算法	431
本章笔记	337	* 26.5 前置重贴标签算法	438
		思考题	446
		本章笔记	449

第六部分 图算法

第 22 章 基本的图算法	341
22.1 图的表示	341
22.2 广度优先搜索	343
22.3 深度优先搜索	349
22.4 拓扑排序	355

第七部分 算法问题选编

第 27 章 多线程算法	453
27.1 动态多线程基础	454
27.2 多线程矩阵乘法	465
27.3 多线程归并排序	468
思考题	472

本章注记	476	思考题	594
第 28 章 矩阵运算	478	本章注记	594
28.1 求解线性方程组	478	第 33 章 计算几何学	595
28.2 矩阵求逆	486	33.1 线段的性质	595
28.3 对称正定矩阵和最小二乘逼近	489	33.2 确定任意一对线段是否相交	599
思考题	493	33.3 寻找凸包	604
本章注记	494	33.4 寻找最近点对	610
第 29 章 线性规划	495	思考题	613
29.1 标准型和松弛型	499	本章注记	615
29.2 将问题表达为线性规划	504	第 34 章 NP 完全性	616
29.3 单纯形算法	507	34.1 多项式时间	619
29.4 对偶性	516	34.2 多项式时间的验证	623
29.5 初始基本可行解	520	34.3 NP 完全性与可归约性	626
思考题	525	34.4 NP 完全性的证明	633
本章注记	526	34.5 NP 完全问题	638
第 30 章 多项式与快速傅里叶变换	527	34.5.1 团问题	638
30.1 多项式的表示	528	34.5.2 顶点覆盖问题	640
30.2 DFT 与 FFT	531	34.5.3 哈密顿回路问题	641
30.3 高效 FFT 实现	536	34.5.4 旅行商问题	644
思考题	539	34.5.5 子集和问题	645
本章注记	541	思考题	647
第 31 章 数论算法	543	本章注记	649
31.1 基础数论概念	543	第 35 章 近似算法	651
31.2 最大公约数	547	35.1 顶点覆盖问题	652
31.3 模运算	550	35.2 旅行商问题	654
31.4 求解模线性方程	554	35.2.1 满足三角不等式的旅行商问题	654
31.5 中国余数定理	556	35.2.2 一般旅行商问题	656
31.6 元素的幂	558	35.3 集合覆盖问题	658
31.7 RSA 公钥加密系统	561	35.4 随机化和线性规划	661
* 31.8 素数的测试	565	35.5 子集和问题	663
* 31.9 整数的因子分解	571	思考题	667
思考题	574	本章注记	669
本章注记	576	第八部分 附录：数学基础知识	
第 32 章 字符串匹配	577	附录 A 求和	672
32.1 朴素字符串匹配算法	578	A.1 求和公式及其性质	672
32.2 Rabin-Karp 算法	580	A.2 确定求和时间的界	674
32.3 利用有限自动机进行字符串匹配	583	思考题	678
* 32.4 Knuth-Morris-Pratt 算法	588	附录注记	678

附录 B 集合等离散数学内容	679	C.2 概率	696
B.1 集合	679	C.3 离散随机变量	700
B.2 关系	682	C.4 几何分布与二项分布	702
B.3 函数	683	* C.5 二项分布的尾部	705
B.4 图	685	思考题	708
B.5 树	687	附录注记	708
B.5.1 自由树	688	附录 D 矩阵	709
B.5.2 有根树和有序树	689	D.1 矩阵与矩阵运算	709
B.5.3 二叉树和位置树	690	D.2 矩阵的基本性质	712
思考题	691	思考题	714
附录注记	692	附录注记	715
附录 C 计数与概率	693	参考文献	716
C.1 计数	693	索引	732

基础知识

这一部分将引导读者开始思考算法的设计和分析问题，简单介绍算法的表达方法、将在本书中用到的一些设计策略，以及算法分析中用到的许多基本思想。本书后面的内容都是建立在这些基础知识之上的。

第 1 章是对算法及其在现代计算系统中地位的一个综述。本章给出了算法的定义和一些算法的例子。此外，本章还说明了算法是一项技术，就像快速的硬件、图形用户界面、面向对象系统和网络一样。

在第 2 章中，我们给出了书中的第一批算法，它们解决的是对 n 个数进行排序的问题。这些算法是用一种伪代码形式给出的，这种伪代码尽管不能直接翻译为任何常规的程序设计语言，但是足够清晰地表达了算法的结构，以便任何一位能力比较强的程序员都能用自己选择的语言将算法实现出来。我们分析的排序算法是插入排序，它采用了一种增量式的做法；另外还分析了归并排序，它采用了一种递归技术，称为“分治法”。尽管这两种算法所需的运行时间都随 n 的值而增长，但增长的速度是不同的。我们在第 2 章分析了这两种算法的运行时间，并给出了一种有用的表示方法来表达这些运行时间。

第 3 章给出了这种表示法的准确定义，称为渐近表示。在第 3 章的一开始，首先定义几种渐近符号，它们主要用于表示算法运行时间的上界和下界。第 3 章余下的部分主要给出了一些数学表示方法。这一部分的作用更多的是为了确保读者所用的记号能与本书的记号体系相匹配，而不是教授新的数学概念。

第 4 章更深入地讨论了第 2 章引入的分治法，给出了更多分治法的例子，包括用于两方阵相乘的 Strassen 方法。第 4 章包含了求解递归式的方法。递归式用于描述递归算法的运行时间。“主方法”是一种功能很强的技术，通常用于解决分治算法中出现的递归式。虽然第 4 章中的相当一部分内容都是在证明主方法的正确性，但是如果跳过这一部分证明内容，也没有什么太大的影响。

第5章介绍概率分析和随机化算法。概率分析一般用于确定一些算法的运行时间，在这些算法中，由于同一规模的不同输入可能有着内在的概率分布，因而在这些不同输入之下，算法的运行时间可能有所不同。在有些情况下，我们假定算法的输入服从某种已知的概率分布，于是，算法的运行时间就是在所有可能的输入之下，运行时间的平均值。在其他情况下，概率分布不是来自于输入，而是来自于算法执行过程中所做出的随机选择。如果一个算法的行为不仅由其输入决定，还要由一个随机数生成器生成的值来决定，那么它就是一个随机化算法。我们可以利用随机化算法强行使算法的输入服从某种概率分布，从而确保不会有某一输入会始终导致算法的性能变坏；或者，对于那些允许产生不正确结果的算法，甚至能够将其错误率限制在某个范围之内。

附录A~D包含了一些数学知识，它们对读者阅读本书可能会有所帮助。在阅读本书之前，读者很有可能已经知道了附录中给出的大部分知识(我们采用的某些符号约定与读者过去见过的可能会有所不同)，因而可以将附录视为参考材料。另外，你很可能从未见过第一部分中给出的内容。第一部分中的所有各章和附录都是以一种入门指南的风格来编写的。

算法在计算中的作用

什么是算法？为什么算法值得研究？相对于计算机中使用的其他技术来说算法的作用是什么？本章我们将回答这些问题。

1.1 算法

非形式地说，**算法**(algorithm)就是任何良定义的计算过程，该过程取某个值或值的集合作为输入并产生某个值或值的集合作为输出。这样算法就是把输入转换成输出的计算步骤的一个序列。

我们也可以把算法看成是用于求解良说明的**计算问题**的工具。一般来说，问题陈述说明了期望的输入/输出关系。算法则描述一个特定的计算过程来实现该输入/输出关系。

例如，我们可能需要把一个数列排成非递减序。实际上，这个问题经常出现，并且为引入许多标准的设计技术和分析工具提供了足够的理由。下面是我们关于**排序问题**的形式定义。

输入： n 个数的一个序列 $\langle a_1, a_2, \dots, a_n \rangle$ 。

输出：输入序列的一个排列 $\langle a'_1, a'_2, \dots, a'_n \rangle$ ，满足 $a'_1 \leq a'_2 \leq \dots \leq a'_n$ 。

例如，给定输入序列 $\langle 31, 41, 59, 26, 41, 58 \rangle$ ，排序算法将返回序列 $\langle 26, 31, 41, 41, 58, 59 \rangle$ 作为输出。这样的输入序列称为排序问题的一个**实例**(instance)。一般来说，**问题实例**由计算该问题解所必需的(满足问题陈述中强加的各种约束的)输入组成。

5

因为许多程序使用排序作为一个中间步，所以排序是计算机科学中的一个基本操作。因此，已有许多好的排序算法供我们任意使用。对于给定应用，哪个算法最好依赖于以下因素：将被排序的项数、这些项已被稍微排序的程度、关于项值的可能限制、计算机的体系结构，以及将使用的存储设备的种类(主存、磁盘或者磁带)。

若对每个输入实例算法都以正确的输出停机，则称该算法是**正确的**，并称正确的算法解决了给定的计算问题。不正确的算法对某些输入实例可能根本不停机，也可能以不正确的回答停机。与人们期望的相反，不正确的算法只要其错误率可控有时可能是有用的。在第 31 章，当我们研究大素数算法时，将看到一个具有可控错误率的算法例子。但是通常我们只关心正确的算法。

算法可以用英语说明，也可以说明成计算机程序，甚至说明成硬件设计。唯一的要求是这个说明必须精确描述所要遵循的计算过程。

算法解决哪种问题

排序绝不是已开发算法的唯一计算问题(当看到本书的厚度时，你可能觉得算法也同样多)。算法的实际应用无处不在，包括以下例子：

- 人类基因工程已经取得重大进展，其目标是识别人类 DNA 中的所有 10 万个基因，确定构成人类 DNA 的 30 亿个化学基对的序列，在数据库中存储这类信息并为数据分析开发工具。这些工作都需要复杂的算法。虽然对涉及的各种问题的求解超出了本书的范围，但是求解这些生物问题的许多方法采用了本书多章内容的思想，从而使得科学家能够有效地使用资源以完成任务。因为可以从实验技术中提取更多的信息，所以既能节省人和机器的时间又能节省金钱。
- 互联网使得全世界的人都能快速地访问与检索大量信息。借助于一些聪明的算法，互联网上的网站能够管理和处理这些海量数据。必须使用算法的问题示例包括为数据传输寻找好的路由(求解这些问题的技术在第 24 章给出)，使用一个搜索引擎来快速地找到特定

6