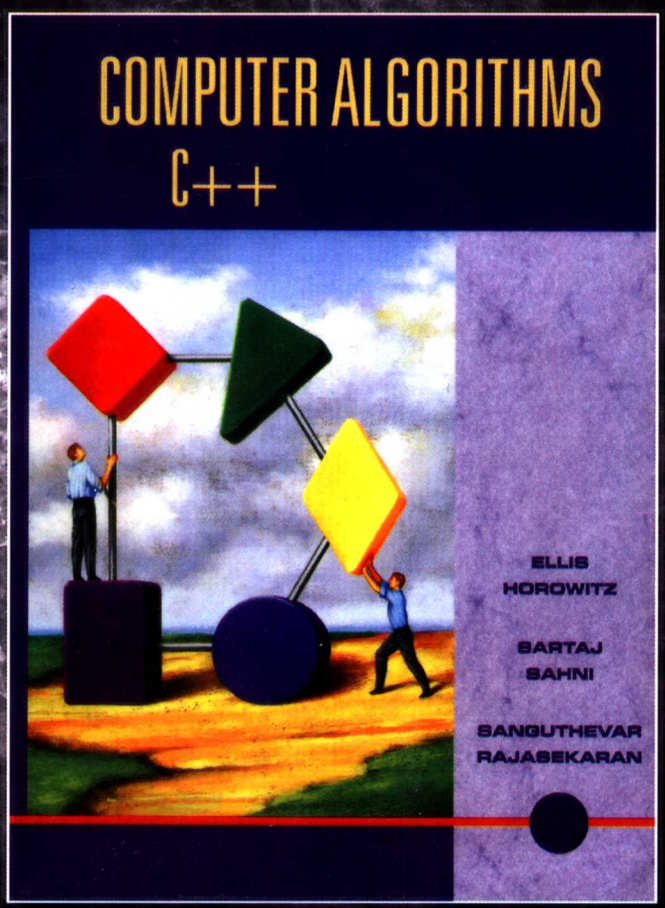
计 算 机 科 学 丛 书

计算机算法（C++版）

(美) Ellis Horowitz 南加州大学 Sartaj Sahni 佛罗里达大学 Sanguthevar Rajasekaran 著 冯博琴 叶茂 高海昌 等译



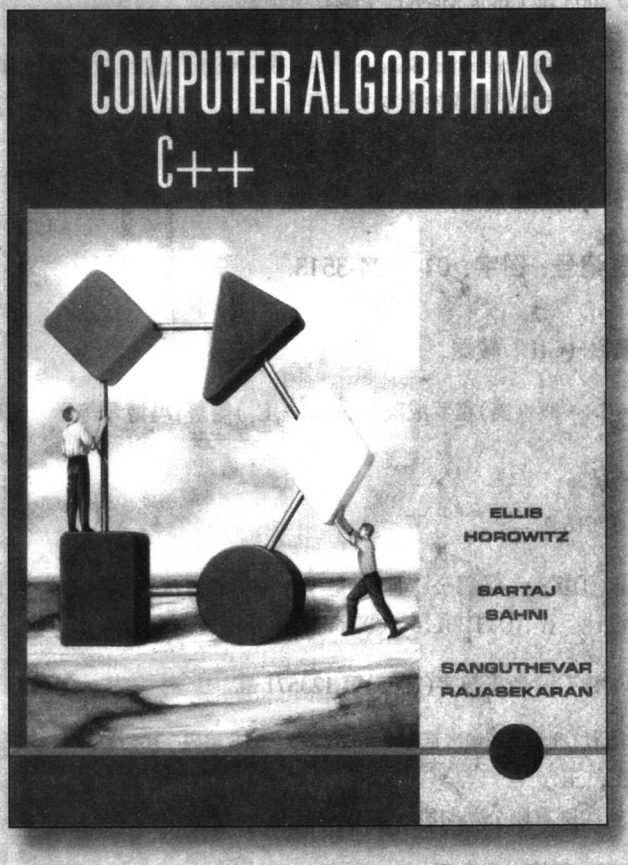
Computer Algorithms, C++

机械工业出版社
China Machine Press

计 算 机 科 学 丛 书

计算机算法
（C++版）

(美) Ellis Horowitz 南加州大学 Sartaj Sahni 佛罗里达大学 Sanguthevar Rajasekaran 著 冯博琴 叶茂 高海昌 等译



Computer Algorithms, C++

机械工业出版社
China Machine Press

本书是计算机算法在设计与分析方面的一本经典著作。书中介绍了算法和算法性能的基本知识，基本的数据结构知识，重点讨论了不同的算法设计策略，研究了下界理论等，提供了计算机算法的设计技术和有效的算法分析，以及大量的详细实例和实际应用。同时，对NP难和NP完全问题能否有效求解进行了分析。本书还汇聚了各种随机算法与并行算法的充分比较。

本书为读者提供了当前流行的对象设计语言C++的实现版本，适合作为高等院校计算机专业的教材，也是计算机算法方面的重要参考书。

Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran: Computer Algorithms/C++ (ISBN: 0-7167-8315-0)

本书版权登记号：图字：01-2004-3513

图书在版编目（CIP）数据

计算机算法：C++版/（美）霍罗威茨（Horowitz, E.）等著；冯博琴等译. －北京：机械工业出版社，2006.1
（计算机科学丛书）
ISBN 7-111-17616-2

Ⅰ.计… Ⅱ.①霍… ②冯… Ⅲ.①电子计算机－算法设计 ②电子计算机－算法分析 ③C语言－程序设计 Ⅳ.①TP301.6 ②TP312

中国版本图书馆CIP数据核字（2005）第123571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　玉
北京诚信伟业印刷有限公司印刷·新华书店北京发行所发行

2006年1月第1版第1次印刷
787mm×1092mm 1/16·29.25印张
印数：0 001-4 000册
定价：55.00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

译 者 序

如果没有算法，计算机就不能完成指定的任务，不能像今天这样为生活带来各种便利，更不要谈模仿人的行为、代替人去工作。不仅计算机专业的从业人员必须了解和使用算法，许多非计算机专业的相关人员也需要设计算法以完成相应的任务。如今算法已广泛应用于社会生活的各个方面，例如进行科学计算、数据分析、知识发现、股票分析、医疗诊断、电梯控制、航班调度、电子邮件系统和病毒防护等。可以毫不夸张地说，计算机上运行的各个程序都是算法的实现。

算法是指令的有限集合，包含了执行任务的过程和思想。算法的设计并不简单，它需要一些策略和技术，好的策略和技术能帮助你以更少的时间设计更好的算法。如果不了解这些基本策略，那么在算法设计的道路上可能会困难重重，甚至完全理不清头绪。此外，算法在设计完成后还需要进行分析。解决某个问题的方法可能有很多，不同的方法会产生不同的算法，需要使用分析技术评估算法的性能、比较各种算法的优劣、论证在实际应用中是否可行。而这一点往往容易被忽略。本书就将教你如何设计并分析算法。

本书第 1 章介绍了算法和算法性能分析的基础知识，这是设计和分析算法必须具备的。第 2 章提供了基本的数据结构知识，为后续章节的介绍做准备。第 3 章～第 9 章是本书的重点，讨论不同的算法设计策略。第 10 章研究下界理论，论证所得算法是不是求解某个问题的渐进最好算法，帮助确定是否存在更快的算法。第 11 章研究 NP 难和 NP 完全问题，讨论了多种经典的 NP 难问题。第 12 章介绍近似算法，提供了求解一部分 NP 难问题的方案。第 13 章～第 15 章讨论并行算法，其中涉及到很多随机算法。

与其他介绍算法的书籍相比，本书有以下 4 个特点。第 1 个特点是对设计技术的重视，作者围绕算法设计的一些基本设计策略组织全书，对这些策略的掌握是快速设计算法的关键。这里介绍的策略并不多，读者在学习本书后便可熟练掌握这些技术。第 2 个特点是算法的涉及面广，尤其是广泛地覆盖了随机算法领域。第 3 个特点是将算法设计和分析结合在一起，帮助读者在学会设计的同时学会分析，用分析来论证设计所得算法的优劣，系统地提高算法设计和分析水平。第 4 个特点是使用 C++ 或伪 C++ 形式给出本书中讨论的很多算法；其中的很多程序都是可执行的，并已经过测试，而伪代码形式的算法则很容易转变成代码实现。

本书很适合作为高校教材使用，作者在前言中给出了按不同的学时和教学要求制定的教学进度，包括作业的布置和测试的安排等。在每个章节后面附有大量习题。另外，本书也很适合作为算法设计与分析的自学用书或参考书。

本书由冯博琴教授组织翻译，叶茂、高海昌和王宇参与译稿、统稿和校阅，参与本书翻译的还有薛亮、韩冰、傅向华、何明、陈宁、朱玉惠、李燕、饶元、霍华及王自强。

在翻译过程中，我们力求忠实、准确地把握原著的内容，但由于译者水平有限，翻译时间仓促，书中难免有错误和不准确之处，敬请广大读者批评指正。

译 者

于西安交通大学

2005 年 6 月

前　言

在计算机科学领域，影响深远的成就之一要算是对算法（algorithm）这个概念的精确定义。自人类发明了能实现基本算术运算的机器以来，就一直在研究可以计算什么和如何做得更好。计算机的出现促进了许多重要算法和设计方法的发现。计算机科学学科已经将算法研究纳为其中的一部分。本书的目的就是将与算法有关的知识连贯地组织起来，帮助学生和研究人员学会自己设计和分析新的算法。

若想用一本书涵盖所有的算法，那么这本书将会非常庞大。以前的算法书籍专注于在深度上研究问题的一个比较窄的领域。对于每个指定的问题提出并分析最有效的算法。这种方法有一个很大的不足，即尽管学生会知道一些快速算法并可能掌握分析工具，但却仍不知道如何设计好的算法。

这些图书缺失的部分是不重视设计（design）技术。设计方面的知识肯定会帮助我们创建好的算法，虽然没有分析工具也不可能确定算法的质量。算法设计和算法分析的教学必须合拍，这为本书的设计提供了参考：围绕算法设计的基本策略组织本书。基本设计策略非常少，而所有算法都可以纳入这些类中进行研究。例如，归并排序和快速排序是分治策略的完美例子，Kruskal 最小生成树算法和 Dijkstra 单源最短路径算法是贪心策略的直接例子。了解这些策略是快速掌握设计技术的第一步。

尽管我们认为对设计和分析的并重是组织算法学习的合适途径，但还必须注意顺序。首先，我们没有给出所有已知的设计原则。例如，线性规划是一种非常成熟的技术，但通常在其课程中讨论。其次，学生应该避免生搬硬套算法设计的规则，每个算法都不一定是从某种单个技术得出的。

第 3～9 章作为本书的重点，讨论不同的设计策略。首先，使用通用术语描述每种策略。通常，如果应用某种策略，就使用“程序概要”（program abstraction）给出计算的框架。然后使用一些例子帮助了解这些策略的复杂性和变形。例子的复杂度逐渐加深，复杂度增加的方式可能有这么几种。通常，我们会给出非常容易理解的问题，只需要一维数组，不需要其他数据结构。通常设计策略都能很明显地为该问题生成正确解。接下来的例子可能需要证明基于这种设计策略的算法是正确的，或者该算法需要更复杂的数据结构（例如，树或图）且分析过程更复杂。这样组织的目的是突出整体的工艺性和算法分析。另一个目的是让学生了解好的程序结构和算法正确性的依据。

第 1～12 章的算法用 C++ 或 C++ 伪代码书写。其中的很多程序都是可执行的，并已经过测试。选择使用 C++ 描述算法的原因是其具有面向对象的特性。基于此以及其他一些原因，C++ 已在计算机科学领域得到广泛应用。本书中的大部分算法都很短小，用于描述算法的语言结构也很简单易懂，所以采用 C++ 并不会妨碍对 C++ 不熟悉的人使用本书。第 13 章、第 14 章和第 15 章讨论并行计算。并行计算领域发展很快，因此在该领域不存在广泛接受的单个模型或某种语言。正因为这个原因，我们使用伪代码表示并行算法。在第 1～12 章中也使用了伪代码描述一些复杂的算法。因为我们觉得这些算法背后的概念用伪代码

要比用晦涩的C++结构来描述更好解释清楚。将这些伪代码转化成C++程序的工作留给读者作为习题。

本书的另一个特征是广泛覆盖了随机算法领域。在第13章、第14章和第15章中讨论的许多算法都是随机的。在其他章节中也提出了一些随机算法。可以使用一个季度的课程介绍第13章、第14章和第15章的随机算法，还可以涉及一小部分附加材料。

有些章节标了星号*，比较适合在高级课程中使用。本书比较适合供大三、大四学生或研究生在一个学期或两个季度内完成。需要有高级语言编程的预备知识，其他所需具备的知识在本书中都已包含。坦率地说，对于已具备成熟编程经验的学生，如果了解数据结构方面的知识对本书的学习会更有帮助。如果学校按季度组织教学，可以在第1个季度讲解第3章～第9章的基本设计技术：分治、贪心算法、动态规划、查找和遍历、回溯、分支限界和代数方法，具体见表1。第2个季度讲解第10章～第15章：下界理论、$\mathcal{NP}$完全和近似算法、PRAM算法、网格算法和超立方体算法，具体见表2。

表1 第1个季度

周 次	主 题	进 度
1	导论	1.1节～1.3节
2	导论、数据结构	1.4节、2.1节、2.2节
3	数据结构	2.3节～2.6节
4	分治算法	第3章，第1次作业
5	贪心算法	第4章，第1次测试
6	动态规划	第5章
7	查找和遍历技术	第6章，第2次作业
8	回溯	第7章
9	分支限界	第8章
10	代数方法	第9章，第3次作业，第2次测试

表2 第2个季度

周 次	主 题	进 度
1	下界理论	10.1节～10.3节
2	下界理论，$\mathcal{NP}$难和$\mathcal{NP}$完全问题	10.4节、11.1节、11.2节
3	$\mathcal{NP}$难和$\mathcal{NP}$完全问题	11.3节、11.4节
4	$\mathcal{NP}$难和$\mathcal{NP}$完全问题、近似算法	11.5节、11.6节、12.1节、12.2节，第1次作业
5	近似算法	12.3节～12.6节，第1次测试
6	PRAM算法	13.1节～13.4节
7	PRAM算法	13.5节～13.9节，第2次作业
8	网格算法	14.1节～14.5节
9	网格算法、超立方体算法	14.6节～14.8节、15.1节～15.3节
10	超立方体算法	15.4节～15.8节，第3次作业，第2次测试

对于按学期安排课程，如果学生不具备数据结构和 O 符号的知识，就学习第 1～7 章、第 11 章和第 13 章，具体见表 3。

如果速度更快些，可以学习第 1～7 章、第 11 章、第 13 章和第 14 章，具体见表 4。

对于有数据结构和 O 符号预备知识的学生，可以开设高级班，学习第 3～11 章、第 13～15 章，具体见表 5。

本书中大部分算法的程序可以从网址 http: //www.cise.ufl.edu/～raj/BOOK.html 得到。请把您的意见发到 raj@cise.ufl.edu。

每章后面有大量习题，可以留作作业。我们发现，最有效的做法是让学生在同一个数据集上执行两个程序并分别计时。本书中的大部分算法都提供了实现细节，所以可以很容易加以使用。将这些 C++ 程序转换成其他语言也很容易。留下的问题就是设计合适的数据集并编写主程序来输出计时结果。测得的时间应该与算法已有的渐近分析结果一致。这是一项重要的工作，兼顾指导性和趣味性。更重要的是，它重视通常容易被忽视的对算法的实践。

表 3　学期－中速（不具备基础知识）

周　次	主　题	进　度
1	导论	1.1 节～1.3 节
2	导论、数据结构	1.4 节、2.1 节、2.2 节
3	数据结构	2.3 节～2.6 节
4	分治算法	3.1 节～3.4 节，第 1 次作业
5	分治算法	3.5 节～3.7 节，第 1 次测试
6	贪心算法	4.1 节～4.4 节
7	贪心算法	4.5 节～4.7 节，第 2 次作业
8	动态规划	5.1 节～5.5 节
9	动态规划	5.6 节～5.10 节
10	查找和遍历技术	6.1 节～6.4 节，第 3 次作业，第 2 次测试
11	回溯	7.1 节～7.3 节
12	回溯	7.4 节～7.6 节
13	$\mathcal{NP}$难和$\mathcal{NP}$完全问题	11.1 节～11.3 节，第 4 次作业
14	$\mathcal{NP}$难和$\mathcal{NP}$完全问题	11.4 节～11.6 节
15	PRAM 算法	13.1 节～13.4 节
16	PRAM 算法	13.5 节～13.9 节，第 5 次作业，第 3 次测试

表 4　学期－快速（不具备基础知识）

周　次	主　题	进　度
1	导论	1.1 节～1.3 节
2	导论、数据结构	1.4 节、2.1 节、2.2 节
3	数据结构	2.3 节～2.6 节
4	分治算法	3.1 节～3.5 节，第 1 次作业
5	分治算法、贪心算法	3.6 节、3.7 节、4.1 节～4.3 节，第 1 次测试
6	贪心算法	4.4 节～4.7 节
7	动态规划	5.1 节～5.7 节，第 2 次作业
8	动态规划、查找和遍历技术	5.8 节～5.10 节、6.1 节、6.2 节

（续）

周　次	主　题	进　度
9	查找和遍历技术、回溯	6.3节、6.4节、7.1节、7.2节
10	回溯	7.3节～7.6节，第3次作业，第2次测试
11	$\mathcal{NP}$难和$\mathcal{NP}$完全问题	11.1节～11.3节
12	$\mathcal{NP}$难和$\mathcal{NP}$完全问题	11.4节～11.6节
13	PRAM算法	13.1节～13.4节，第4次作业
14	PRAM算法	13.5节～13.9节
15	网格算法	14.1节～14.3节
16	网格算法	14.4节～14.8节，第5次作业，第3次测试

表5　学期－高级班（快速）

周　次	主　题	进　度
1	分治算法	3.1节～3.5节
2	分治算法、贪心算法	3.6节、3.7节、4.1节～4.3节
3	贪心算法	4.4节～4.7节
4	动态规划	第5章，第1次作业
5	查找和遍历技术	第6章，第1次测试
6	回溯	第7章
7	分支限界	第8章，第2次作业
8	代数方法	第9章
9	下界理论	第10章
10	$\mathcal{NP}$难和$\mathcal{NP}$完全问题	11.1节～11.3节，第3次作业，第2次测试
11	$\mathcal{NP}$难和$\mathcal{NP}$完全问题	11.4节～11.6节
12	PRAM算法	13.1节～13.4节
13	PRAM算法	13.5节～13.9节，第4次作业
14	网格算法	14.1节～14.5节
15	网格算法、超立方体算法	14.6节～14.8节、15.1节～15.3节
16	超立方体算法	15.4节～15.8节，第5次作业，第3次测试

致谢

非常感谢 Martin J. Biernat，Jeff Jenness，Saleem Khan，Ming－Yang Kao，Douglas M. Campbell 和 Stephen P. Leach 提供的重要意见，这使本书有了极大的提高。感谢 UF 的学生指出初稿中的一些错误。还要感谢 Teo Gonzalez，Danny Krizanc 和 David Wei 仔细阅读了本书的部分章节。

Ellis Horowitz
Sartaj Sahni
Sanguthevar Rajasekaran

目　录

第1章 导 论

1.1 什么是算法

“算法”一词源于波斯作家Abu Ja'far Mohammed ibn Musa al Khowarizmi 的名字(大约公元 825 年)，他写了一本关于数学的教科书。在计算机科学中，“算法”具有特殊的意义，特指计算机用来解决某一问题的方法。这正是“算法”一词与“过程”、“技术”或“方法”等词的不同之处。

定义 1.1 [算法]：算法是完成特定任务的有限指令集。所有的算法必须满足下面的标准：

1. **输入**。由外部提供零个或多个输入量。
2. **输出**。至少产生一个输出量。
3. **明确性**。每条指令必须清楚，不具模糊性。
4. **有限性**。如果跟踪算法的指令，那么对于所有的情况，算法经过有限步以后终止。
5. **有效性**。每条指令必须非常基础，原则上使用笔和纸就可以实现。每个操作不仅要满足标准 3 要求的明确性，而且必须可行。

算法由有限步组成，每一步需要一个或多个操作。因为计算机可能会执行这些操作，所以对算法包含的操作类型必须存在一些约束条件。

标准 1 和 2 要求算法产生一个或多个输出，并且有零个或多个由外部提供的输入。根据标准 3，每个操作必须是明确的，即，应当做什么必须非常清楚。例如，因为指令“将 6 或 7 与 x 相加”或“计算 5/0”，对于应该执行两个可能中的哪一个，或结果是什么并不清楚，所以是不允许的。

本书中关于算法的第 4 个标准是算法必须在有限次的操作后终止。与其相关的终止时间应该比较短。例如，设计一个算法，用来确定国际象棋比赛中一个给定位置是否是赢棋位置。该算法考察从开始位置起，所有可能的走棋路线和对手所有可能的对抗手段。其困难之处在于即使使用最现代的计算机，也可能需要运算上亿年的时间才能作出判定。因此必须关注每个算法的效率分析。

标准 5 要求每个操作必须是有效的。至少在原理上，每一步必须可以由人用笔和纸在有限的时间内完成。执行整数算术运算是一个有效操作，但实数的算术运算就不是，因为某些值只能用无限小数来表示。对这样的两个数进行加法操作会违背有效性标准。

明确的和有效的算法又被称为可计算过程。可计算过程的一个重要例子是数字计算机的操作系统。该过程用于控制作业的执行，遵循如下方式：当无作业可用时，并不终止，而是以等待状态继续运行，直到输入一个新的作业。虽然可计算过程包括了这类重要例子，但本书仅限于研究能终止的可计算过程。

为满足明确性这一标准，可以利用一种程序设计语言来描述算法。通过设计，让这些语言中的每个合法的句子都具有唯一的意思。程序是对算法使用程序设计语言的表示。有时过程、函数和子程序在程序中作为同义词使用。本书的大多数读者可能都有编程和在计算机上

运行算法的经验，这是比较理想的。因为在研究一般的概念之前，如果具有一些相关实际经验将非常有帮助。也许你在开始设计问题的初始解决方案时会碰到一些困难，或者不能确定两个算法当中哪一个更好，本书的目的就是教你怎样作出决策。

算法的研究包括许多重要和活跃的领域，可以分为四个不同的领域：

2 1. *怎样设计算法*——创建算法是一门艺术，可能永远不可能完全自动地设计算法。本书的主要目标是研究不同的设计技术，这些技术对创建好的算法经证明非常有用。掌握了这些设计策略之后，设计新的和有用的算法会变得更容易。本书的很多章节内容根据我们认为的主要算法设计方法进行组织。读者可能现在就希望浏览一下目录表，了解这些方法的名称。这些技术中的一些可能已广为人知，一些已被证明非常有用，因而本书也进行了介绍。动态规划就是这样的一种技术。还有一些技术不仅在计算机科学领域，在其他领域也特别有用，例如在运筹学或电子工程中。本书只给出了关于算法形式化的许多方法的介绍。所有这些方法在不同的领域中有着广泛的应用，包括计算机科学在内。但一些重要的设计技术，如线性、非线性以及整数规划并不包含在本书中，它们包含在其他传统课程当中。

2. *怎样验证算法*——设计好一个算法以后，必须证明它对所有可能的合法输入可以计算出正确的结果。这一过程被称为*算法验证*。算法不必一定要表示成程序的形式，以任何精确的方式表述就足够了。验证的目标是确保算法可以独立于最终使用的编程语言正确运行。证明方法的有效性以后，就可以编写程序，开始第二个阶段。第二阶段称为*程序证明*，有时被称为*程序验证*。问题的解以两种形式表述来进行正确性的证明。一种形式通常为程序，由关于程序输入和输出变量的断言进行注释，这些断言通常用谓词演算来表示。第二种形式称为*规范*，也可以用谓词演算来表示。这两种形式是等价的，因为对于任意给定的合法输入，它们给出相同的输出。程序正确性的完全证明要求程序设计语言的每条语句必须精确定义，并且必须证明所有基本操作都正确。所有这些细节可能产生比程序长得多的证明。

3. *怎样分析算法*——这一研究领域被称为算法分析。当算法执行时，使用计算机的中央处理器(CPU)执行操作，用存储器(包括直接或辅助存取器)来存放程序和数据。*算法分析*或*性能分析*是确定算法需要的计算时间和存储空间。这是一个需要数学技能有挑战意义的领域。这个
3 研究的一个重要结果是通过算法分析能定量地比较两个算法，另外还能预测软件是否满足所有存在的有效约束。通常要回答算法在最好情况下、最坏情况下或者平均情况下的表现之类的问题。对于本书中的每个算法都给出了相应的分析。1.3.2 节更完整地叙述了分析过程。

4. *怎样测试程序*——测试程序由两个步骤组成：调试和评测(或称为性能度量)。在样本数据集合上执行程序，确定是否产生错误结果。如果产生就改正程序，这个过程就是*调试*。然而，E. Dijkstra 已经指出，“调试只能指出存在的错误，但是不能证明没有错误。”在不能确认在样本数据上的输出是否正确的情况下，可以遵从下面的策略：让多个程序设计人员为同一个问题开发程序，然后比较这些程序产生的输出结果。如果输出结果一致，那么它们很可能正确。正确性的证明比成千上万个测试更有价值(若证明过程是正确的)，因为这保证程序对所有可能的输入都是正确的。*评测*或者*性能度量*是在数据集合上执行正确的程序并度量计算出结果所需要花费的时间和空间。这些时间特性很有用，它们可以确认前面所作的分析并指出可以实现算法优化的逻辑位置。有关时间复杂度度量的描述可以在 1.3.5 节中找到。对此处的一些算法，给出设计数据集大小的方法，以帮助调试和评测。

这 4 类用于概括与本书中的算法相关的问题。因为不能期望完全覆盖所有这些主题，所