



国外经典教材·计算机科学与技术



Introduction to
The Design and Analysis
of Algorithms

算法设计与分析基础

(美) Anany Levitin 著
潘彦 译



清华大学出版社

国外经典教材·计算机科学与技术

算法设计与分析基础

(美) Anany Levitin 著

潘彦 译

清华大学出版社

北京

内 容 简 介

作者基于丰富的教学经验,开发了一套对算法进行分类的新方法。这套方法站在通用问题求解策略的高度,对现有的大多数算法都能进行很好的分类,从而使本书的读者能够沿着一条清晰的、一致的、连贯的道路来探索算法设计与分析这一迷人领域。

本书十分适合计算机专业的本科高年级学生或者研究生学习。另外,由于本书的介绍深入浅出,只要具备数据结构和离散数学的知识,任何有兴趣探究算法秘密的读者也可以自学本书。

Simplified Chinese edition copyright © 2004 by PEARSON EDUCATION ASIA LIMITED and TSINGHUA UNIVERSITY PRESS.

Original English language title from Proprietor's edition of the Work.

Original English language title: Introduction to The Design and Analysis of Algorithms, 1st Edition by Anany Levitin, Copyright © 2003

EISBN: 0-201-74395-7

All Rights Reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Pearson Education, Inc.

This edition is authorized for sale only in the People's Republic of China (excluding the Special Administrative Region of Hong Kong and Macao).

本书中文简体翻译版由 Pearson Education 授权给清华大学出版社在中国境内(不包括中国香港、澳门特别行政区)出版发行。

北京市版权局著作权合同登记号 图字: 01-2003-1782

本书封面贴有 Pearson Education(培生教育出版集团)激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

算法设计与分析基础/ (美) 莱维丁(Levitin, A.) 著; 潘彦译. —北京: 清华大学出版社, 2004

(国外经典教材·计算机科学与技术)

书名原文: Introduction to The Design and Analysis of Algorithms

ISBN 7-302-08656-7

I. 算… II. ①莱… ②潘… III. ①算法设计—高等学校—教材 ②算法分析—高等学校—教材 IV. TP301.6

中国版本图书馆CIP数据核字(2004)第046190号

出版者: 清华大学出版社 地 址: 北京清华大学学研大厦

<http://www.tup.com.cn> 邮 编: 100084

社总机: 010-62770175 客户服务: 010-62776969

文稿编辑: 汤涌涛 葛昊晗

封面设计: 久久度文化

印刷者: 北京季峰印刷有限公司

装订者: 三河市新茂装订有限公司

发行者: 新华书店总店北京发行所

开 本: 185×260 印张: 25.25 字数: 614千字

版 次: 2004年6月第1版 2004年6月第1次印刷

书 号: ISBN 7-302-08656-7/TP·6224

印 数: 1~4500

定 价: 45.00元

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770175-3103或(010)62795704。

出版说明

近年来，我国的高等教育特别是计算机学科教育，进行了一系列大的调整和改革，急需一批门类齐全、具有国际先进水平的计算机经典教材，以适应当前我国计算机科学的教學需要。通过使用国外先进的经典教材，可以了解并吸收国际先进的教学思想和教学方法，使我国的计算机科学教育能够跟上国际计算机教育发展的步伐，从而培育出更多具有国际水准的计算机专业人才，增强我国计算机产业的核心竞争力。为此，我们从国外知名的出版集团 Pearson 引进这套“国外经典教材·计算机科学与技术”教材。

作为全球最大的图书出版机构，Pearson 在高等教育领域有着不凡的表现，其下属的 Prentice Hall 和 Addison Wesley 出版社是全球计算机高等教育的龙头出版机构。清华大学出版社与 Pearson 出版集团长期保持着紧密友好的合作关系，这次引进的“国外经典教材·计算机科学与技术”教材大部分出自 Prentice Hall 和 Addison Wesley 两家出版社。为了组织该套教材的出版，我们在国内聘请了一批知名的专家和教授，成立了一个专门的教材编审委员会。

教材编审委员会的运作从教材的选题阶段即开始启动，各位委员根据国内外高等院校计算机科学及相关专业的现有课程体系，并结合各个专业的培养方向，从 Pearson 出版的计算机系列教材中精心挑选针对性强的题材，以保证该套教材的优秀性和领先性，避免出现“低质重复引进”或“高质消化不良”的现象。

为了保证出版质量，我们为这套教材配备了一批经验丰富的编辑、排版、校对人员，制定了更加严格的出版流程。本套教材的译者，全部来自于对应专业的高校教师或拥有相关经验的 IT 专家。每本教材的责编在翻译伊始，就定期不间断地与该书的译者进行交流与反馈。为了尽可能地保留与发扬教材原著的精华，在经过翻译、排版和传统的三审三校之后，我们还请编审委员或相关的专家教授对文稿进行审读，以最大程度地弥补和修正在前面一系列加工过程中对教材造成的误差和瑕疵。

由于时间紧迫和受全体制作人员自身能力所限，该套教材在出版过程中很可能还存在一些遗憾，欢迎广大师生来电来信批评指正。同时，也欢迎读者朋友积极向我们推荐各类优秀的国外计算机教材，共同为我国高等院校计算机教育事业贡献力量。

清华大学出版社

2004.03.20

国外经典教材·计算机科学与技术

编 审 委 员 会

主任委员：

孙家广 清华大学教授

副主任委员：

周立柱 清华大学教授

委员（按姓氏笔画排序）：

王成山	天津大学教授
王 珊	中国人民大学教授
冯少荣	厦门大学教授
冯全源	西南交通大学教授
刘乐善	华中科技大学教授
刘腾红	中南财经政法大学教授
吉根林	南京师范大学教授
孙吉贵	吉林大学教授
阮秋琦	北京交通大学教授
何 晨	上海交通大学教授
吴百锋	复旦大学教授
李 彤	云南大学教授
沈钧毅	西安交通大学教授
邵志清	华东理工大学教授
陈 纯	浙江大学教授
陈 钟	北京大学教授
陈道蓄	南京大学教授
周伯生	北京航空航天大学教授
孟祥旭	山东大学教授
姚淑珍	北京航空航天大学教授
徐佩霞	中国科学技术大学教授
徐晓飞	哈尔滨工业大学教授
秦小麟	南京航空航天大学教授
钱培德	苏州大学教授
曹元大	北京理工大学教授
龚声蓉	苏州大学教授
谢希仁	中国人民解放军理工大学教授

译者序

关于这本书的意义和它的主要内容，作者在序和跋中讲得已经够多够好的了。我只想谈谈自己在翻译过程中的一些感想。

首先，我要恭喜拿到这本书的读者，你们是幸运的。我们读书的年代，很少有大学会教授算法课，惟一和这门课程比较相近的是数据结构。这一点我求证过很多人，只是遇到一个例外，一位毕业于西南师范大学的同事告诉我，她的母校同时开设了数据结构课程和算法课程。遗憾的是，十年过去了，她很难回忆起这两门课程之间的显著区别。很明显，如果算法设计课程真的像本书所描述得那么精彩的话，我们很有必要对算法设计及分析课程给予高度的重视。之所以这么说不是因为数据结构不重要，而是因为，算法设计与分析的角度和高度都是不同于数据结构的：

- 从考虑问题的角度来看，数据结构主要关心不同的数据结构在计算机解题中的作用和效率；而算法设计与分析则关心不同的算法设计技术的适用性和效率。这使得这两门课程在某种程度上有所重叠，但又无法相互替代。
- 从考虑问题的高度来看，数据结构关心的是计算机解题的具体问题，所以很适合作为计算机技术的入门课程之一；但算法设计与分析则要宽泛地多，它很适合计算机专业的学生学习，但又不仅限于解决计算机解题的问题。本书的作者说得好，“授人以鱼，不如授人以渔”（Learning such techniques is akin to learning to fish as opposed to being given a fish caught by somebody else）。当把算法设计与分析提升为一种解决问题的通用方法时，读者一定会有很大的收获，而且有可能终生受益。

其次，我要庆幸自己有机会翻译一本这么好的图书。这不仅因为在翻译的过程当中，我为自己补上了一门算法设计与分析课程，而且，能够普及优秀的图书也是我的愿望。惟一的担心是自己的能力不足，所以书中无法避免会存在种种失误。读者应该及时向我指出，以便在未来的版本中不断完善。

最后，我要感谢清华大学出版社对我的信任，以及在翻译过程中，诸位编辑给予的帮助。还要感谢我的家人，他们都大力支持我的工作，使我能够以饱满的热情完成翻译工作。尤其是我的太太李靓，她辅助我翻译了本书的大部分章节，并且输入了所有的公式，我很感激她的时刻陪伴。

潘彦，2003年11月19日于上海
panyan@dhzq.com

前 言

一个人接受科技教育得到的最大收获，是那些能够受用一生的一般性智能工具^①。

——George Forsythe, 《计算机科学家到来以前我们做什么》，1968

无论是计算科学还是计算实践，算法都在其中扮演着重要角色。由于这一事实，这门学科中出现了大量的教科书。它们在介绍算法的时候，基本上都选择了以下两种方案中的一种。第一种方案是按照问题的类型对算法分类。这类教科书安排了不同的章节分别讨论排序、查找、图等算法。这种做法的优点是，对于解决同一问题的不同算法，它能够立即比较这些算法的效率。其缺点在于，由于过于强调问题的类型，它忽略了对算法设计技术的讨论。

第二种方案围绕着算法设计技术来组织章节。在这种结构中，即使算法来自于不同的计算领域，如果它们采用了相同的设计技术，就会被编成一组。从各方获得的信心（例如[BaY95]）使我相信，这种结构对算法设计与分析的基础课程尤为适合。强调算法设计技术有3个重要原因。第一，学生们在解决新问题时，可以运用这些技术设计出新的算法。从实用的角度看，这使学习算法设计技术成为一种很有价值的努力。第二，学生们会尝试着按照算法的内在设计方法，来对已知的众多算法进行分类。计算机科学教育的一个主要目的，就是使学生们通过学习，能够了解不同应用领域的算法间的共性。毕竟，每门科学都会倾向于把它的重要主题归纳为几个甚至一个理论。第三，依我看来，算法设计技术作为问题求解的一般性策略，在解决计算机领域以外的问题时，也能发挥相当大的作用。

目前已经出版了一些围绕算法设计技术进行组织的教科书（参见[BB96]，[HSR98]，[NN98]）。但这些书的问题在于，它们未加批判地遵循了相同的设计技术分类法。无论是从理论还是从教育的观点来看，这种分类法都存在一些严重的缺陷。其中最显著的缺陷就是无法对许多重要的算法进行分类。由于这种局限性，这些书的作者不得不在按照设计技术进行分类的同时，另外增加一些章节，来讨论特殊的问题类型。遗憾的是，这种改变导致课程缺乏一致性，而且很可能会使学生产生疑惑。

算法设计技术的新分类法

现有的算法设计技术分类法的缺陷给我带来了困难，它激发我开发一个基于设计技术

^① 译注：George Forsythe 认为，在这些工具当中，最重要的3项依次是自然语言、数学和计算机科学。

的新分类法[Lev99]，该分类法就是本书的基础。以下是这个新分类法的几个主要优势：

- 新分类法比传统分类法更容易理解。它包含的某些设计策略，比如蛮力法、减治法、变治法和时空权衡，几乎从不被人们当作重要的设计范例。
- 新分类法很自然地覆盖了许多传统方法无法分类的经典算法（欧几里得算法、堆排序、查找树、散列法、拓扑排序、高斯消去法、霍纳法则等，不胜枚举）。所以，它能够以一种连贯、一致的方式表达这些经典算法的标准内容。
- 新分类法很自然地包容了一些设计技术的重要变种（比如：它涵盖了减治法的 3 个变种和变治法的 3 个变种）。
- 在分析算法效率时，新分类法与分析方法结合得更好（参见附录 B）。

设计技术作为问题求解的一般性策略

本书中，设计技术主要应用于计算机科学中的经典问题。这里惟一的创新是引入了一些数值算法的内容，这些算法也包含在相同的通用框架之中。（“计算机教育大纲 2001——一种全新模式的计算机科学课程体系”[CC01]中提倡包含数值算法）但我们也可以把这些设计技术看成问题求解的一般性工具，它们的应用不仅限于传统的计算问题和数学问题。有两个因素令这一点变得尤其重要。第一，越来越多的计算应用超越了它们传统的领域，并且有足够理由使人相信，这种趋势会愈演愈烈。第二，人们渐渐认识到，提高学生们的求解能力是高等教育的一个主要目标。为了满足这个目标，在计算机科学课程体系中，安排一门关于算法设计和分析的课程是非常合适的，因为它会告诉学生如何应用一些特定的策略来解决问题。虽然我并不建议将算法设计和分析课程变成一门教授一般性问题求解方法的课程，但我的确认为，我们不应错过算法设计和分析课程提供的这样一个独一无二的机会。所以，本书包含的某些应用是和谜题相关的。虽然利用谜题来教授算法课程决不是一个新的想法，但本书打算通过引进一些全新的例子来系统地实现这个思路。

如何使用本书

我的目标是写一本既不泛泛而谈，又能被学生们独立阅读的教材。为了实现这个目标，本书做了如下努力：

- 根据 George Forsythe 的观点（参见引语），我试图着重强调那些隐藏在算法设计和分析背后的主要思想。在选择特定的算法来阐述这些思想的时候，我并不倾向于涉及大量的算法，而是选择那些能够揭示出一个根本的设计技术或是分析方法的算法。幸运的是，大多数经典算法满足这个要求。
- 第 2 章主要分析算法的效率，该章将分析非递归算法的方法和分析递归算法的方法区别开来。这一章还花了一些篇幅介绍算法实证分析以及算法可视化。
- 书中系统地为读者安排了一些问题。其中有些问题是经过精心设计的，而且立即

提供答案，目的是引起读者的注意或是疑问。其余问题的用意是防止读者走马观花，而对本书的理解不能达到令人满意的水平。

- 每一章结束时都会对本章最重要的概念和结论做一个总结。
- 本书包含大约 600 道习题。有些习题是用于练习；另外一些则是为了指出书中正文部分所涉及内容的重要意义，或是为了介绍一些书中没有涉及到的算法。有一些习题是从因特网上找到的，还有些习题是让读者为后面章节将要涉及的内容做准备的。较难的习题数量不多，会在教师用书中用一种特殊的记号标注出来（因为有些学生可能没有勇气做那些标有难度的习题，所以本书没有对习题标注难度）。谜题类的习题用了一种特殊的图标做标注。
- 本书所有的习题都附有提示。除了编程练习，习题的详细解法都能够在教师用书中找到，符合条件的读者可以从出版社得到教师用书（请联系 Addison-Wesley 公司的销售代表，或者发电子邮件至 aw.cse@aw.com）。本书的任何读者都可以向 www.aw.com/cssupport 网站寻求支持。

读者所需的知识背景

本书假定读者已经学习了离散数学的标准课程和一门基础性的编程课程。在这样一种知识背景下，读者应该能够掌握本书的内容而不会遇到太大的困难。尽管如此，1.4 节、附录 A 和附录 B 仍然会对基本的数据结构、必须用到的求和公式和递推关系分别做了复习和回顾。只有 3 个小节（2.2 节、10.4 节和 11.4 节）会用到一些简单的微积分知识；如果学生们缺少必要的微积分知识，完全可以跳过这 3 个包含微积分内容的小节，这样做不会妨碍对本书其余部分的理解。

进度安排

如果打算开设一门围绕算法设计技术来讲解算法设计和分析理论的基础课程，本书可以作为这门课的教科书。对于一门典型的单学期课程来说，本书涵盖的内容可能过于丰富了。大体上来说，跳过第 3 章至第 11 章的部分内容不会影响读者对后面部分的理解。本书的任何一个部分都可以安排学生自学。特别要指出的是，2.6 节和 2.7 节分别介绍了实证分析和算法可视化，这两小节的内容可以结合练习^①布置给学生。

下面提供一种针对一个学期课程的教学安排，它是按照 40 课时的集中授课时间设计的。

^① 译注：“练习”的原文为“project”，一般应该翻译成“项目”，但国外一般将布置在课后完成的、较大型的、要求实际演练的习题称为 project，国内没有相应的称呼，所以姑且译为“练习”。

课次	主题	小节
1、2	课程简介	1.1-1.3
3、4	分析框架: O 、 Θ 和 Ω 符号	2.1, 2.2
5	非递归算法的数学分析	2.3
6、7	递归算法的数学分析	2.4, 2.5 (+附录 B)
8	蛮力算法	3.1, 3.2 (+3.3)
9	穷举查找	3.4
10-12	分治法: 合并排序、快速排序、折半查找	4.1-4.3
13	分治法的其他例子	4.4 或 4.5 或 4.6
14-16	减一算法: 插入排序、DFS 和 BFS、拓扑排序	5.1-5.3
17	减常量算法	5.5
18	减变量算法	5.6
19-21	实例化简、预排序、高斯消去法、平衡查找树	6.1-6.3
22	改变表现: 堆和堆排序	6.4
23	改变表现: 霍纳法则和二进制幂	6.5
24	问题化简	6.6
25-27	时空权衡: 串匹配、散列法、B 树	7.2-7.4
28-30	动态规划算法	8.1-8.4 (选 3 节)
31-33	贪婪算法: Prim 算法、Kruskal 算法、Dijkstra 算法、哈夫曼算法	9.1-9.4
34	下界的参数	10.1
35	决策树	10.2
36	P 、 NP 和 NP 完全问题	10.3
37	数值算法	10.4 (+11.4)
38	回溯	11.1
39	分支界限	11.2
40	NP 困难问题的近似算法	11.3

Anany Levitin

anany.levitin@villanova.edu

2002 年 8 月

目 录

第1章 绪论	1
1.1 算法的概念	2
习题 1.1	6
1.2 算法问题求解基础	7
习题 1.2	13
1.3 重要的问题类型	15
习题 1.3	18
1.4 基本数据结构	20
习题 1.4	30
小结	31
第2章 算法效率分析基础	33
2.1 分析框架	34
习题 2.1	40
2.2 渐进符号和基本效率类型	41
习题 2.2	47
2.3 非递归算法的数学分析	49
习题 2.3	53
2.4 递归算法的数学分析	55
习题 2.4	61
2.5 例题：斐波那契数列	63
习题 2.5	67
2.6 算法的经验分析	68
习题 2.6	72
2.7 算法可视法	74
小结	77
第3章 蛮力法	79
3.1 选择排序和冒泡排序	80
习题 3.1	83
3.2 顺序查找和蛮力字符串匹配	83
习题 3.2	85
3.3 最近对和凸包问题的蛮力算法	86
习题 3.3	90
3.4 穷举查找	91

习题 3.4	95
小结	96
第 4 章 分治法	99
4.1 合并排序	101
习题 4.1	103
4.2 快速排序	104
习题 4.2	108
4.3 折半查找	109
习题 4.3	111
4.4 二叉树遍历及其相关特性	112
习题 4.4	114
4.5 大整数乘法和 Strassen 矩阵乘法	115
习题 4.5	119
4.6 用分治法解最近对问题和凸包问题	120
习题 4.6	123
小结	124
第 5 章 减治法	125
5.1 插入排序	127
习题 5.1	129
5.2 深度优先查找和广度优先查找	130
习题 5.2	135
5.3 拓扑排序	137
习题 5.3	139
5.4 生成组合对象的算法	141
习题 5.4	144
5.5 减常因子算法	145
习题 5.5	148
5.6 减可变规模算法	149
习题 5.6	153
小结	154
第 6 章 变治法	155
6.1 预排序	156
习题 6.1	158
6.2 高斯消去法	159
习题 6.2	166
6.3 平衡查找树	168
习题 6.3	174
6.4 堆和堆排序	175
习题 6.4	180
6.5 霍纳法则和二进制幂	181
习题 6.5	186

6.6 问题化简	187
习题 6.6	193
小结	194
第 7 章 时空权衡	197
7.1 计数排序	198
习题 7.1	200
7.2 串匹配中的输入增强技术	201
习题 7.2	209
7.3 散列法	210
习题 7.3	214
7.4 B 树	215
习题 7.4	218
小结	219
第 8 章 动态规划	221
8.1 计算二项式系数	222
习题 8.1	224
8.2 Warshall 算法和 Floyd 算法	225
习题 8.2	231
8.3 最优二叉查找树	232
习题 8.3	236
8.4 背包问题和记忆功能	237
习题 8.4	240
小结	241
第 9 章 贪婪技术	243
9.1 Prim 算法	244
习题 9.1	248
9.2 Kruskal 算法	249
习题 9.2	255
9.3 Dijkstra 算法	256
习题 9.3	259
9.4 哈夫曼树	260
习题 9.4	263
小结	264
第 10 章 算法能力的极限	265
10.1 如何求下界	266
习题 10.1	270
10.2 决策树	271
习题 10.2	275
10.3 P、NP 和 NP 完全问题	276
习题 10.3	282

10.4 数值算法的挑战	284
习题 10.4	291
小结	292
第 11 章 超越算法能力的极限	295
11.1 回溯	295
习题 11.1	301
11.2 分支界限	302
习题 11.2	309
11.3 NP 困难问题的近似算法	310
习题 11.3	317
11.4 解非线性方程的算法	319
习题 11.4	326
小结	327
跋	329
附录 A: 算法分析的实用公式	333
对数的性质	333
组合学	333
重要的求和公式	333
求和乘法法则	334
用定积分逼近求和式	334
向下取整和向上取整公式	334
其他	335
附录 B: 递推关系简明指南	337
序列和递推关系	337
递推关系的求解方法	338
算法分析中的常见递推类型	341
习题提示	349
第 1 章	349
第 2 章	351
第 3 章	355
第 4 章	357
第 5 章	360
第 6 章	364
第 7 章	367
第 8 章	369
第 9 章	371
第 10 章	373
第 11 章	376
参考文献	381

第1章 绪 论

有两种思想，像珠宝商放在天鹅绒上的宝石一样熠熠生辉，一个是微积分，另一个就是算法。微积分以及在微积分基础上建立起来的数学分析体系造就了现代科学，而算法则造就了现代世界。

——David Berlinski, 《算法的出现》，2000

为什么要学习算法？对于一个即将从事计算机专业的人士来说，无论从理论还是从实践的角度，学习算法都是有必要的。从实践的角度来看，我们必须了解计算领域中不同问题的一系列标准算法；此外，我们还要具备设计新算法和分析其效率的能力。从理论的角度来看，对算法的研究（有时称为“**算法学**”）已经被公认为是计算机科学的基石。David Harel 写了一本非常好的书，他直截了当地将其命名为《算法：计算的灵魂》。书中是这样阐述这个问题的：

算法不仅是计算机科学的一个分支，它更是计算机科学的核心。而且，可以毫不夸张地说，它和绝大多数的科学、商业和技术都是相关的。（参见[Har92]）

即使对于非计算机相关专业的学生来说，学习算法的理由也是非常充分的。坦率地说，没有算法，计算机程序将不复存在。而且，随着计算机应用在我们的工作和生活的方方面面日益变得不可或缺，需要学习算法的人也越来越多。

学习算法的另一个理由是可以用它来开发人们的分析能力。毕竟，算法可以看作是解决问题的一类特殊方法——它虽非问题的答案，但它是经过准确定义的，用来获得答案的过程。因此，无论是否涉及计算机，特定的算法设计技术都能看作是问题求解的有效策略。当然，算法思想天生固有的精确性限制了它能够解决的问题种类。比如说，我们无法找到一个使人生活愉快的算法，也不能找到一个使人富有和出名的算法。但另一方面，从教育角度来看，这种必要的精确性却是很重要的。Donald Knuth，作为算法学历史上最卓越的计算机科学家之一，是这样论述这个问题的：

受过良好训练的计算机科学家知道如何处理算法：如何构造算法、操作算法、理解算法以及分析算法。这些知识远不只是为了编写良好的计算机程序而准备的。算法是一种一般性的智能工具，一定有助于我们对其他学科的理解，不管是化学、语言学、音乐，还是另外的学科。为什么算法会有这种作用呢？我们可以这样理解：人们常说，一个人只有把知识教给别人，才能真正掌握它。实际上，一个人只有把知识教给“计算机”，才能“真正”掌握它，也就是说，将知识表

述为一种算法……比起简单地按照常规去理解事物，用算法将其形式化会使我们获得更加深刻的理解。（参见[Knu96]）

我们从 1.1 节开始着手涉及算法的概念。作为例子，我们将针对同一问题（求最大公约数）使用 3 种不同的算法。这样做有几个理由。首先，无论是谁，在中学时代就应该非常熟悉这个问题了。其次，它揭示了一个重要的观点：不同的算法能够解决相同的问题。非常具有代表性的是，这些算法的解题思路不同、复杂程度不同、解题效率也不相同。第三，其中一个算法应当作为本书介绍的第一个算法，不仅因为它的悠久历史——两千多年前它就出现在了欧几里得的著作中，还因为它不朽的力量和重要性。最后，通过中学里求最大公约数的步骤，使我们能够着重指出每一个算法都必须满足的严格要求。

1.2 节讲述算法问题求解。我们将会讨论有关算法设计和分析的一些重要内容。内容涉及算法问题求解的不同方面，包括问题的分析、如何表述算法来证明其正确性以及算法的效率分析。这一节并不能教给我们一个绝招，来为任意的问题设计算法。这是一个公认的事实——这样的绝招是不存在的。但读者在日后安排自己的算法设计和分析工作时，1.2 节的内容会有用的。

1.3 节讨论几种问题类型，经验证明，无论对于学习算法还是应用算法来说，这些问题类型都是极其重要的。实际上，有些教材就是围绕这些问题类型组织内容的。虽然许多人（包括我们）都认为围绕算法设计技术来组织内容更有优势，但无论如何，了解这些重要的问题类型是十分必要的。这—是因为这些类型是实际应用中最常遇到的，二是由于本书从头到尾都会用它们来演示一些特殊的算法设计技术。

1.4 节对基本的数据结构做了一番回顾。安排这一节的目的与其说是为了讨论这方面的内容，还不如说是为了把它作为读者的参考材料。如果需要了解更详细的内容，有许多关于这个主题的优秀书籍可供参阅，其中大多数都是为某种特定的编程语言编写的。

1.1 算法的概念

什么是算法？虽然没有一个大家公认的用语来描述这个概念，但针对其含义，我们还是有着基本共识的。

算法是一系列解决问题的清晰指令，也就是说，能够对符合一定规范的输入，在有限时间内获得所要求的输出。

这个定义可以用一个简单的图形来说明（图 1.1）。

定义中使用了“指令”这个词，这意味着有人或物能够理解和执行给出的指令，我们将其称为“computer”。请记住，在电子计算机发明以前，“computer”的含义是指那些从事数学计算的人。现在，“computer”当然是特指那些做每件事情都越发不可缺少的，无所不在的电子设备。然而，请注意这一点，虽然绝大多数算法打算最终依靠计算机来执行，但算法的概念本身并不依赖于这样一个假设。

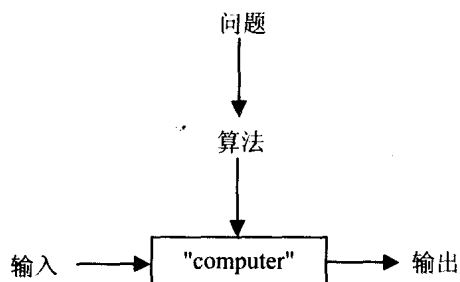


图 1.1 算法的概念

作为阐明算法概念的几个例子，本节会讨论 3 种方法，用于解决同一个问题：计算两个整数的最大公约数。这些例子会帮助我们阐明几个要点：

- 算法的每一个步骤都必须清晰、明确，这是来不得半点含糊的。
- 算法所处理的输入的值域必须仔细定义。
- 同样一种算法可以用几种不同的形式来描述。
- 可能存在几种解决相同问题的算法。
- 针对同一个问题的算法可能会基于完全不同的解题思路，而且解题速度也会有显著不同。

让我们回到例子上来，我们将两个不全为 0 的非负整数 m 和 n 的最大公约数记为 $\text{gcd}(m, n)$ ，代表能够整除（即余数为 0） m 和 n 的最大正整数。亚历山大的欧几里得（公元前 3 世纪）所著的《几何原本》，以系统性论述几何学而著称，在其中的一卷里，他对解决这一问题的算法做了一个简单的介绍。用现代数学术语来表述，欧几里得算法基于的方法是重复应用下列等式，直到 $m \bmod n$ 等于 0：

$$\text{gcd}(m, n) = \text{gcd}(n, m \bmod n) \quad (m \bmod n \text{ 表示 } m \text{ 除以 } n \text{ 之后的余数})$$

因为 $\text{gcd}(m, 0) = m$ （为什么？）， m 最后的取值也就是 m 和 n 的初值的最大公约数。

举例来说， $\text{gcd}(60, 24)$ 可以这样计算：

$$\text{gcd}(60, 24) = \text{gcd}(24, 12) = \text{gcd}(12, 0) = 12$$

如果你对这个算法的印象还不够深的话，可以做做练习 1.1 的第 4 题，试着求一些较大数的最大公约数。

下面是该算法一个更加结构化的描述：

用于计算 $\text{gcd}(m, n)$ 的欧几里得算法

第一步：如果 $n = 0$ ，返回 m 的值作为结果，同时过程结束；否则，进入第二步。

第二步：用 n 去除 m ，将余数赋给 r 。

第三步：将 n 的值赋给 m ，将 r 的值赋给 n ，返回第一步。

我们也可以使用伪代码来描述这个算法：