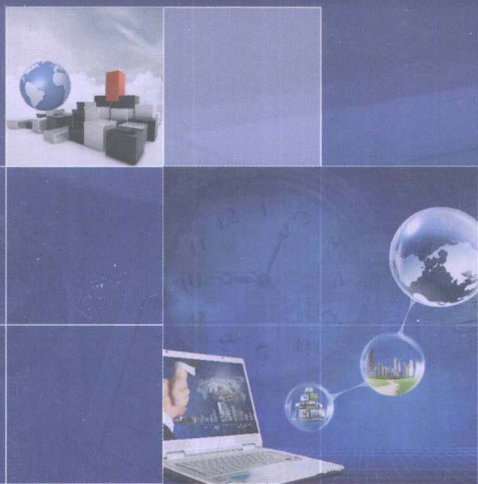


算法设计、分析与应用教程

主 编 李文书 何利力



Algorithm

- 以讲故事的形式将概念和算法的精髓娓娓道来
- 阐述对应算法的理论基础、特点及其解题框架
- 活学活用经典问题，巧妙设计与分析ACM真题

TP301.6
10191

21 世纪

计算机案例型规划教材

算法设计、分析与应用教程

主 编 李文书 何利力
副主编 叶海荣 韩逢庆 董世都
黄 海 茅海军



80038977

国家自然科学基金、省自然科学基金及钱江人才(B类)资助项目
(60702069, Y1080851, 31271102, Y12H290045, 2012R10054)

“海洋机电装备技术”浙江省重中之重学科重点实验室
开放基金资助项目
(1203720-N)

浙江省人事厅留学人员科技活动项目
(1104707-M)

浙江省科协育才工程资助项目
宁波市第四批科技项目

(2011B710007)学
图书馆藏书



北京大学出版社
PEKING UNIVERSITY PRESS

内 容 简 介

为了适应培养 21 世纪计算机人才的需要,结合我国高等院校教育工作的现状,立足培养学生能跟上国际计算机科学技术的发展水平,更新教学内容和教学方法,本书以算法设计策略为知识单元,系统地介绍计算机算法的设计方法与分析技巧,以期信息技术相关学科的学生提供广泛而坚实的算法基础知识。本书主要内容包括算法概述、递归与分治策略、动态规划、贪心算法、回溯算法、分支限界算法、图的搜索算法、加密算法与安全机制、P 和 NP 问题等。书中既涉及经典算法及分析,又包括指导 ACM 时练习的相关算法。

本书内容丰富、观点新颖、理论联系实际,采用 C/C++ 语言描述算法,简明清晰、结构紧凑、可读性强。本书可作为高等院校信息技术相关专业本科生和研究生学习算法设计的教材,也可供广大工程技术人员和自学读者学习参考。

图书在版编目(CIP)数据

算法设计、分析与应用教程/李文书,何利力主编. —北京:北京大学出版社,2014.7

(21 世纪全国应用型本科计算机案例型规划教材)

ISBN 978-7-301-24352-7

I. ①算… II. ①李…②何… III. ①电子计算机—算法设计—高等学校—教材②电子计算机—算法分析—高等学校—教材 IV. ①TP301.6

中国版本图书馆 CIP 数据核字(2014)第 122270 号

书 名: 算法设计、分析与应用教程

著作责任者: 李文书 何利力 主编

策划编辑: 郑 双

责任编辑: 郑 双

标准书号: ISBN 978-7-301-24352-7/TP · 1337

出版发行: 北京大学出版社

地 址: 北京市海淀区成府路 205 号 100871

网 址: <http://www.pup.cn> 新浪官方微博: @北京大学出版社

电子信箱: pup_6@163.com

电 话: 邮购部 62752015 发行部 62750672 编辑部 62750667 出版部 62754962

印 刷 者: 北京富生印刷厂

经 销 者: 新华书店

787 毫米×1092 毫米 16 开本 23.5 印张 彩插 4 536 千字

2014 年 7 月第 1 版 2014 年 7 月第 1 次印刷

定 价: 49.00 元

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究

举报电话: 010-62752024 电子信箱: fd@pup.pku.edu.cn

21 世纪全国应用型本科计算机案例型规划教材

专家编审委员会

(按姓名拼音顺序)

主 任 刘瑞挺

副主任 陈 钟 蒋宗礼

委 员 陈代武 房爱莲 胡巧多 黄贤英

江 红 李 建 娄国焕 马秀峰

祁亨年 汪新民 王联国 谢安俊

解 凯 徐 苏 徐亚平 宣兆成

姚喜妍 于永彦 张荣梅

信息技术的案例型教材建设

(代丛书序)

刘瑞挺

北京大学出版社第六事业部在 2005 年组织编写了《21 世纪全国应用型本科计算机系列实用规划教材》，至今已出版了 50 多种。这些教材出版后，在全国高校引起热烈反响，可谓初战告捷。这使北京大学出版社的计算机教材市场规模迅速扩大，编辑队伍茁壮成长，经济效益明显增强，与各类高校师生的关系更加密切。

2008 年 1 月北京大学出版社第六事业部在北京召开了“21 世纪全国应用型本科计算机案例型教材建设和教学研讨会”。这次会议为编写案例型教材做了深入的探讨和具体的部署，制定了详细的编写目的、丛书特色、内容要求和风格规范。在内容上强调面向应用、能力驱动、精选案例、严把质量；在风格上力求文字精练、脉络清晰、图表明快、版式新颖。这次会议吹响了提高教材质量第二战役的进军号。

案例型教材真能提高教学的质量吗？

是的。著名法国哲学家、数学家勒内·笛卡儿(Rene Descartes, 1596—1650)说得好：“由一个例子的考察，我们可以抽出一条规律。(From the consideration of an example we can form a rule.)”事实上，他发明的直角坐标系，正是通过生活实例而得到的灵感。据说是在 1619 年夏天，笛卡儿因病住进医院。中午他躺在病床上，苦苦思索一个数学问题时，忽然看到天花板上有一只苍蝇飞来飞去。当时天花板是用木条做成正方形的格子。笛卡儿发现，要说出这只苍蝇在天花板上的位置，只需说出苍蝇在天花板上的第几行和第几列。当苍蝇落在第四行、第五列的那个正方形时，可以用(4, 5)来表示这个位置……由此他联想到可用类似的办法来描述一个点在平面上的位置。他高兴地跳下床，喊着“我找到了，找到了”，然而不小心把国际象棋撒了一地。当他的目光落到棋盘上时，又兴奋地一拍大腿：“对，对，就是这个图”。笛卡儿锲而不舍的毅力，苦思冥想的钻研，使他开创了解析几何的新纪元。千百年来，代数与几何，井水不犯河水。17 世纪后，数学突飞猛进的发展，在很大程度上归功于笛卡儿坐标系和解析几何学的创立。

这个故事，听起来与阿基米德在浴缸洗澡而发现浮力原理，牛顿在苹果树下遇到苹果落到头上而发现万有引力定律，确有异曲同工之妙。这就证明，一个好的例子往往能激发灵感，由特殊到一般，联想出普遍的规律，即所谓的“一叶知秋”、“见微知著”的意思。

回顾计算机发明的历史，每一台机器、每一颗芯片、每一种操作系统、每一类编程语言、每一个算法、每一套软件、每一款外部设备，无不像闪光的珍珠串在一起。每个案例都闪烁着智慧的火花，是创新思想不竭的源泉。在计算机科学技术领域，这样的案例就像大海岸边的贝壳，俯拾皆是。

事实上，案例研究(Case Study)是现代科学广泛使用的一种方法。Case 包含的意义很广：包括 Example 例子，Instance 事例、示例，Actual State 实际状况，Circumstance 情况、事件、境遇，甚至 Project 项目、工程等。

我们知道在计算机的科学术语中，很多是直接来自日常生活的。例如 Computer 一词早在 1646 年就出现于古代英文字典中，但当时它的意义不是“计算机”而是“计算工人”，即专门从事简单计算的工人。同理，Printer 当时也是“印刷工人”而不是“打印机”。正是

由于这些“计算工人”和“印刷工人”常出现计算错误和印刷错误，才激发查尔斯·巴贝奇(Charles Babbage, 1791—1871)设计了差分机和分析机，这是最早的专用计算机和通用计算机。这位英国剑桥大学数学教授、机械设计专家、经济学家和哲学家是国际公认的“计算机之父”。

20世纪40年代，人们还用 Calculator 表示计算机。到电子计算机出现后，才用 Computer 表示计算机。此外，硬件(Hardware)和软件(Software)来自销售人员。总线(Bus)就是公共汽车或大巴，故障和排除故障源自格瑞斯·霍普(Grace Hopper, 1906—1992)发现的“飞蛾子”(Bug)和“抓蛾子”或“抓虫子”(Debug)。其他如鼠标、菜单……不胜枚举。至于哲学家进餐问题，理发师睡觉问题更是操作系统文化中脍炙人口的经典。

以计算机为核心的信息技术，从一开始就与应用紧密结合。例如，ENIAC 用于弹道曲线的计算，ARPANET 用于资源共享以及核战争时的可靠通信。即使是非常抽象的图灵机模型，也受益于二战时图灵博士破译纳粹密码工作的关系。

在信息技术中，既有许多成功的案例，也有不少失败的案例；既有先成功而后失败的案例，也有先失败而后成功的案例。好好研究它们的成功经验和失败教训，对于编写案例型教材有重要的意义。

我国正在实现中华民族的伟大复兴，教育是民族振兴的基石。改革开放30年来，我国高等教育在数量上、规模上已有相当的发展。当前的重要任务是提高培养人才的质量，必须从学科知识的灌输转变为素质与能力的培养。应当指出，大学课堂在高新技术的武装下，利用PPT进行的“高速灌输”、“翻页宣科”有愈演愈烈的趋势，我们不能容忍用“技术”绑架教学，而是让教学工作乘信息技术的东风自由地飞翔。

本系列教材的编写，以学生就业所需的专业知识和操作技能为着眼点，在适度的基础知识与理论体系覆盖下，突出应用型、技能型教学的实用性和可操作性，强化案例教学。本套教材将会有机融入大量最新的示例、实例以及操作性较强的案例，力求提高教材的趣味性和实用性，打破传统教材自身知识框架的封闭性，强化实际操作的训练，使本系列教材做到“教师易教，学生乐学，技能实用”。有了广阔的应用背景，再造计算机案例型教材就有了基础。

我相信北京大学出版社在全国各地高校教师的积极支持下，精心设计，严格把关，一定能够建设出一批符合计算机应用型人才培养模式的、以案例型为创新点和兴奋点的精品教材，并且通过一体化设计、实现多种媒体有机结合的立体化教材，为各门计算机课程配齐电子教案、学习指导、习题解答、课程设计等辅导资料。让我们用锲而不舍的毅力，勤奋好学的钻研，向着共同的目标努力吧！

刘瑞挺教授 本系列教材编写指导委员会主任、全国高等院校计算机基础教育研究会副会长、中国计算机学会普及工作委员会顾问、教育部考试中心全国计算机应用技术证书考试委员会副主任、全国计算机等级考试顾问。曾任教育部理科计算机科学教学指导委员会委员、中国计算机学会教育培训委员会副主任。PC Magazine《个人电脑》总编辑、CHIP《新电脑》总顾问、清华大学《计算机教育》总策划。

前 言

算法作为数学的一个分支已经存在几百年了，然而算法真正焕发青春得到长足的发展还是发生在 20 世纪计算机发明的时代。随着计算机技术的广泛应用，人们越来越清楚地认识到，作为计算机科学与工程最主要的技术——程序设计，其灵魂就是解决问题的算法。

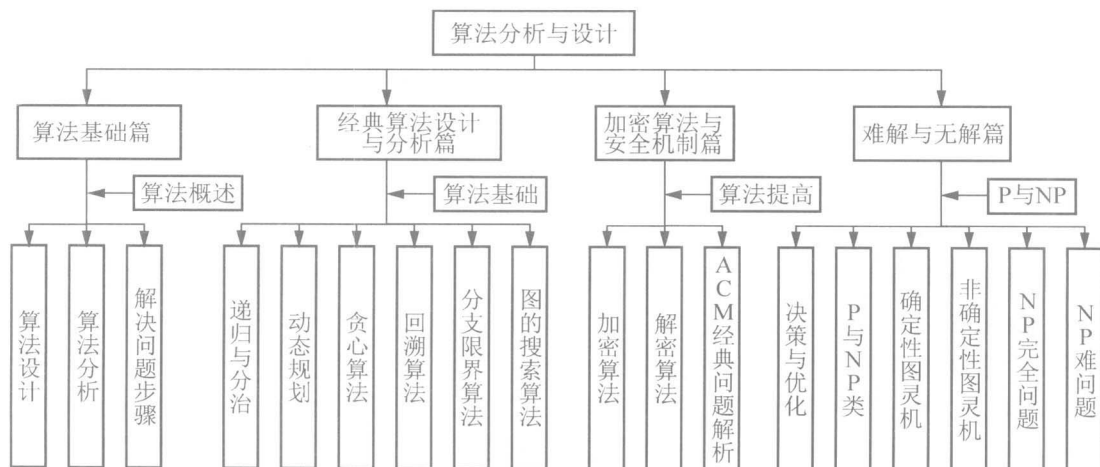
很多人对算法学习都有一种“枯燥繁难”的先入为主的感觉。如何有效地学习算法的设计与分析，是本书试图与读者一起探讨的最重要的目标之一。其实，算法都是针对具体问题的，可以说算法无处不在。每个人每天都在使用不同的算法来活出自己的人生。比如，你去食堂买饭会选择较短的队列，而有人则可能选择一个推进速度更快的队列；每天起床后，你可能先读一会儿书，再去吃早饭，另外一个人则可能先去吃早饭，然后读书。所有这些行为都是算法问题或算法一部分的体现。也许运行这些算法并没在你的思想意识里，也许你并不知道算法在帮助自己的生活，但它确实是存在的。这些算法也许没有经过精心设计，没有经过仔细分析，但它还是算法！

此外，算法的发现问题是相关的问题驱动的。拿排序来说，因为生活中到处都充满次序，每个人都要接受自己在某个次序里的位置，如各种排名、评估、民意调查等，最后的结果都体现为一个次序。看来，“没有次序无以成方圆”并不是空穴来风。而谈到排序用的方法，人们很自然地想到插入法，因为这种朴素的算法和人的思维方式非常类似，它就是人们打牌时整理手中扑克的算法。但随着数据量的增大，插入排序的效率缺陷迅速变为人们无法容忍的缺点。于是人们发明了归并排序、堆排序、快速排序等，这些排序的方法大大改善了速度，但是人们却并不满足于此。因此，又发明了效率更高的线性排序。

因此，本书的特点是以讲故事的形式将概念和算法的精华娓娓道来，然后讨论对应算法的理论基础、特点及其形式表达，最后通过对经典问题和 ACM 问题的设计与分析，让读者由浅入深地理解和消化该章的理论知识。这和我们中国人学习过程的 4 个境界(比、从、北、化)是对应的。其中，“比”就是接受先辈留下的智慧，“从”就是跟随先辈的经验认识世界，“北”就是借鉴先辈的智慧和经验解决现在的问题，而“化”就是融合先辈和自己的经验产生新的知识、解决新的问题。

著名数学家华罗庚先生曾经说过，读数学书若不做习题似“入宝山而空返”。先生的意思是说思想不经过实践检验，再好的理论和技能也难以掌握和应用。为此，本书在每个经典问题和 ACM 的算法分析之后，都配有对应问题实现的代码。

本书内容主要包括四部分，具体如下：



第一部分(第1章)为算法基础篇,介绍了算法的基本概念及算法分析的相关基础知识,包括算法的设计、算法的分析、解决问题的一般步骤。

第二部分(第2~7章)为经典算法设计与分析篇,介绍了递归与分治、动态规划、贪心算法、回溯算法、分支限界算法及图的搜索算法。以故事起,典型算法设计策略为知识单元,采用算法基本思想、算法描述、算法分析的模式展开,从算法设计和算法分析的理论入手,根据各类算法的基本技术原理,给出算法的分析与证明方法,并将经典算法与 ACM 相结合,实现理论与实践相结合。其中,递归与分治主要包括算法的设计、分析及解决问题的一般步骤等分析;动态规划主要包括递归算法、分治算法及相关 ACM 的经典问题解析;贪心算法主要包括理论基础、贪心选择的性质、求解过程及相关 ACM 的经典问题解析;回溯算法主要包括其中问题的解空间、搜索的解空间、回溯的基本步骤及相关的经典问题(图的 m 着色问题、 n 皇后问题、装载问题、0-1 背包问题、旅行商问题、流水作业调度问题等)的解析;分支限界算法主要包括搜索策略、分支结点的选择、限界函数及相关的经典问题(单源最短路径问题、装载问题、0-1 背包问题、旅行商问题、ACM 经典问题等);图的搜索算法主要包括图的广度优先搜索遍历、深度优先搜索遍历、有向图的强连通分支、无向图的又连通分支、流网络与最大流问题及相关的 ACM 经典问题解析。

第三部分(第8章)为加密算法与安全机制篇。主要包括 RSA 公钥密码算法、因子分析算法、离散对数密码算法、离散对数算法及相关 ACM 经典问题解析。

第四部分(第9章)为难解与无解篇。从决策问题、优化问题出发,介绍图灵机、非确定性图灵机,给出 NP 完全性的理论基础及 NP 完全问题、NP 难问题近似算法的基本技术及分析方法。

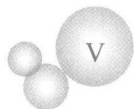
本书由李文书、何利力主编,叶海荣、韩逢庆、董世都、黄海、茅海军为副主编,郑军红、邹玉金、刘建平、徐洋洋、盛实旺、赵超、朱宁馨、王柯、谷广兵、史步娥、陈莲娜、奚圣波、刘智等为参编。具体编写分工如下:第1、2、3、7、9章及附录由李文书教授编写,第4章主要由茅海军副教授编写,第5、6章由韩逢庆教授和董世都副教授编写,第8章由黄海副教授编写,叶海荣、邹玉金及史步娥对各章的 ACM 经典算法进行编写,何利力教授、郑军红博士和刘建平副教授对各章的编写进行了多次指导与修改。此外,徐洋洋、盛实旺、赵超、朱宁馨、王柯和谷广兵等参加校对书稿、算法实验、课件制作、

试题演算等工作，在此一并对他们表示感谢！在写作过程中作者得到了许多教授的帮助和支持，审稿时提出了许多宝贵意见，在此表示衷心的感谢！同时，在本书的编写过程中，得到了北京大学出版社有关同志的关心和大力支持，谨此一并表示衷心的感谢。

由于编者水平有限，加之时间仓促，书中难免存在不妥之处，敬请读者指正。本书的程序代码可通过编者电子信箱获得，E-mail 地址：wshlee@163.com。也可以从教学资源网 www.pup6.com 下载。

编 者

2014年2月





非淡泊无以明志，非宁静无以致远

目 录

第 1 章 算法概述.....1	2.3.5 TOYS(难度:★★☆☆☆).....53
1.1 引言.....1	2.3.6 Cable master (难度:★★☆☆☆).....56
1.1.1 算法的描述.....2	2.4 小结.....58
1.1.2 算法的特性.....2	2.5 习题.....59
1.1.3 为什么学习算法.....3	第 3 章 动态规划.....62
1.2 算法的设计.....5	3.1 何谓动态规划.....63
1.3 算法的分析.....8	3.1.1 动态规划的基本思想.....63
1.3.1 正确性分析.....9	3.1.2 设计动态规划法的步骤.....63
1.3.2 时空效率分析.....10	3.1.3 动态规划问题的特征.....63
1.3.3 时空特性分析.....13	3.1.4 动态规划与静态规划的关系...64
1.4 解决问题的一般步骤.....13	3.2 矩阵连乘积问题.....65
1.5 小结.....15	3.2.1 分析最优解的结构.....67
1.6 习题.....16	3.2.2 建立递归关系.....68
第 2 章 递归与分治策略.....17	3.2.3 计算最优值.....69
2.1 递归算法.....18	3.2.4 构造最优解.....71
2.1.1 递归的概念.....18	3.3 动态规划算法的基本要素.....72
2.1.2 具有递归特性的问题.....19	3.3.1 最优子结构.....72
2.1.3 递归算法分析.....22	3.3.2 重叠子问题.....72
2.2 分治策略.....28	3.3.3 备忘录方法.....73
2.2.1 分治法的基本步骤.....28	3.4 最长公共子序列.....75
2.2.2 分治法的适用条件.....29	3.4.1 最长公共子序列的结构.....75
2.2.3 二分搜索技术.....29	3.4.2 子问题的递归结构.....76
2.2.4 棋盘覆盖问题.....30	3.4.3 计算最优值.....76
2.2.5 快速排序.....33	3.4.4 构造最长公共子序列.....78
2.2.6 大整数乘法.....36	3.5 最大子段和.....78
2.2.7 矩阵乘法.....39	3.5.1 递归关系分析.....78
2.3 ACM 经典问题解析.....45	3.5.2 算法实现.....79
2.3.1 蜂窝问题 (难度:★☆☆☆☆).....45	3.6 0-1 背包问题.....80
2.3.2 Humble Numbers (难度:★★☆☆☆).....46	3.6.1 递归关系分析.....81
2.3.3 Copying Books (难度:★★★☆☆).....48	3.6.2 算法实现.....81
2.3.4 Fractal(难度:★★★☆☆).....51	3.7 ACM 经典问题解析.....83
	3.7.1 数塔(难度:★★☆☆☆).....83
	3.7.2 免费馅饼 (难度:★★★☆☆).....84

3.7.3	Dividing (难度: ★★★☆☆).....	86	4.7.4	算法实现.....	121
3.7.4	Win the Bonus (难度: ★★★★★).....	88	4.7.5	复杂度分析.....	122
3.7.5	Monkey and Banana (难度: ★★★★★).....	90	4.8	ACM 经典问题解析.....	122
3.7.6	Railroad(难度: ★★★★★).....	93	4.8.1	Fat Mouse Trade (难度: ★★☆☆☆).....	122
3.8	小结.....	96	4.8.2	Sorting the Photos (难度: ★★★★★).....	124
3.9	习题.....	97	4.8.3	Moving Tables (难度: ★★★★★).....	126
第 4 章	贪心算法	101	4.8.4	Box of Bricks (难度: ★★★★★).....	127
4.1	活动安排问题.....	102	4.8.5	Wooden Sticks (难度: ★★★★★).....	128
4.2	贪心算法的理论基础.....	104	4.8.6	钓鱼问题 (难度: ★★★★★).....	130
4.2.1	贪心算法的基本思想.....	105	4.8.7	树形 DP 问题 (难度: ★★★★★).....	133
4.2.2	贪心算法的基本要素.....	105	4.8.8	Frogs' Neighborhood (难度: ★★★★★).....	135
4.2.3	贪心算法的基本步骤.....	106	4.9	小结.....	137
4.3	删数问题.....	107	4.10	习题.....	138
4.3.1	贪心策略选择.....	107	第 5 章	回溯法	140
4.3.2	最优子结构.....	107	5.1	回溯法的基本思想.....	140
4.3.3	算法实现.....	107	5.1.1	问题的解空间.....	141
4.3.4	复杂度分析.....	108	5.1.2	搜索的解空间.....	143
4.4	背包问题.....	109	5.1.3	回溯的基本步骤.....	144
4.4.1	最优子结构性质.....	109	5.1.4	回溯法实现.....	145
4.4.2	贪心选择性质.....	110	5.2	图的 m 着色问题.....	147
4.4.3	算法实现.....	110	5.2.1	问题的解空间.....	147
4.4.4	复杂度分析.....	112	5.2.2	约束条件.....	148
4.5	最优装载问题.....	112	5.2.3	搜索解空间.....	148
4.5.1	贪心选择性质.....	113	5.2.4	代码实现.....	148
4.5.2	最优子结构性质.....	113	5.2.5	算法时间复杂度分析.....	150
4.5.3	算法实现.....	113	5.3	n 皇后问题.....	150
4.5.4	复杂度分析.....	114	5.3.1	问题的解空间.....	151
4.6	单源最短路径.....	115	5.3.2	约束条件.....	151
4.6.1	算法基本思想.....	115	5.3.3	搜索过程.....	151
4.6.2	贪心选择性质.....	116	5.3.4	算法的时间复杂度分析.....	154
4.6.3	最优子结构性质.....	117	5.4	装载问题.....	154
4.6.4	Dijkstra 算法实现.....	117	5.4.1	问题的解空间.....	154
4.6.5	复杂度分析.....	119			
4.7	多处最优服务次序问题.....	120			
4.7.1	贪心选择策略.....	120			
4.7.2	贪心选择性质.....	120			
4.7.3	最优子结构性质.....	120			

5.4.2	约束条件.....	154	6.2.2	算法描述与设计.....	186
5.4.3	限界条件.....	154	6.2.3	算法实现.....	188
5.4.4	搜索过程.....	155	6.3	装载问题.....	190
5.4.5	算法效率分析.....	157	6.3.1	问题描述.....	190
5.5	0-1 背包问题.....	157	6.3.2	算法设计与实现.....	191
5.5.1	问题的解空间.....	157	6.4	0-1 背包问题.....	196
5.5.2	约束条件.....	157	6.4.1	问题描述.....	196
5.5.3	限界条件.....	157	6.4.2	算法描述与设计.....	196
5.5.4	搜索过程.....	158	6.4.3	算法实现.....	198
5.5.5	算法效率分析.....	160	6.5	旅行商问题.....	202
5.6	旅行商问题.....	160	6.5.1	问题描述.....	202
5.6.1	问题的解空间.....	161	6.5.2	算法描述与设计.....	203
5.6.2	约束条件.....	161	6.5.3	算法实现.....	204
5.6.3	限界条件.....	161	6.7	ACM 经典问题.....	209
5.6.4	搜索解空间.....	161	6.7.1	布线问题	
5.6.5	时间复杂度分析.....	163		(难度: ★★★☆☆).....	209
5.7	批处理流水作业调度问题.....	163	6.7.2	方格调整问题	
5.7.1	问题的解空间.....	163		(难度: ★★★☆☆).....	212
5.7.2	约束条件.....	164	6.7.3	旅行售货员问题	
5.7.3	限界条件.....	164		(难度: ★★★☆☆).....	213
5.7.4	搜索过程.....	164	6.7.4	Grandpa's Estate	
5.7.5	时间复杂度分析.....	166		(难度: ★★★☆☆).....	216
5.8	ACM 经典问题解析.....	166	6.7.5	Find The Multiple	
5.8.1	Dreisam Equations			(难度: ★★★☆☆).....	218
	(难度: ★★★☆☆).....	166	6.8	小结.....	220
5.8.2	A Plug for UNIX		6.9	习题.....	220
	(难度: ★★★☆☆).....	170	第 7 章	图的搜索算法.....	222
5.8.3	回文构词检测(Anagram Checker)		7.1	图的广度优先搜索遍历.....	224
	(难度: ★★☆☆☆).....	174	7.1.1	算法描述与分析.....	224
5.8.4	Unshuffle		7.1.2	程序实现.....	227
	(难度: ★★★☆☆).....	178	7.2	图的深度优先搜索遍历.....	232
5.9	小结.....	181	7.2.1	算法描述与分析.....	232
5.10	习题.....	181	7.2.2	程序实现.....	234
第 6 章	分支限界算法.....	183	7.2.3	有向无圈图的拓扑排序.....	237
6.1	分支限界法的基本理论.....	184	7.3	有向图的强连通分支.....	244
6.1.1	分支限界法的搜索策略.....	184	7.3.1	算法描述与分析.....	244
6.1.2	分支节点的选择.....	185	7.3.2	程序实现.....	247
6.1.3	限界函数.....	185	7.4	无向图的双连通分支.....	250
6.2	单源最短路径问题.....	186	7.4.1	算法描述与分析.....	250
6.2.1	问题描述.....	186	7.4.2	程序实现.....	254

7.5	流网络与最大流问题	256	8.7	习题	303
7.5.1	算法描述与分析	256	第 9 章 P 和 NP 问题浅析		304
7.5.2	程序实现	263	9.1	决策问题和优化问题	305
7.6	ACM 经典问题解析	265	9.2	何谓 P 类和 NP 类问题	306
7.6.1	Is It A Tree? (难度: ★★★☆☆)	265	9.2.1	P 类问题	306
7.6.2	Stockbroker Grapevine (难度: ★★★☆☆)	267	9.2.2	NP 类问题	307
7.6.3	A Plug for UNIX (难度: ★★★☆☆)	269	9.3	(确定性)图灵机	307
7.7	小结	273	9.3.1	图灵机的定义	307
7.8	习题	273	9.3.2	k 带图灵机形式化描述	308
第 8 章 公钥加密算法		281	9.3.3	图灵机计算实例	308
8.1	RSA 公钥密码算法	283	9.4	非确定性图灵机	311
8.1.1	算法描述	283	9.4.1	非确定性图灵机定义	311
8.1.2	快速模幂算法	284	9.4.2	非确定性图灵机形式化 描述	312
8.1.3	素数的生成	285	9.4.3	非确定性图灵机计算实例	312
8.1.4	扩展欧几里得算法	288	9.4.4	非确定性算法	313
8.2	因子分解算法	290	9.4.5	NP 类问题的定义	314
8.2.1	Pollard's $p-1$ 法	290	9.4.6	NP 难(NP-hard)	315
8.2.2	Pollard's rho 法	291	9.5	NP 完全问题 P^*	315
8.3	离散对数密码算法	293	9.5.1	定义	316
8.3.1	Diffie-Hellman 密钥交换 协议	293	9.5.2	多项式时间规约	316
8.3.2	ElGamal 公钥密码算法	294	9.5.3	库克定理	318
8.4	离散对数算法	295	9.5.4	3-SAT 问题	320
8.4.1	小步/大步法	295	9.5.5	NP 完全问题的近似算法	321
8.4.2	Pohlig-Hellman 法	297	9.6	NP 难问题的近似算法*	332
8.5	ACM 的经典问题	299	9.6.1	旅行商问题的近似算法	333
8.5.1	简单的加密算法 (难度: ★★☆☆☆)	299	9.6.2	背包问题的近似算法	339
8.5.2	古代密码 (难度: ★★★☆☆)	300	9.7	小结	342
8.6	小结	302	9.8	习题	343
			附录 A 求和		345
			附录 B 数论入门		352
			参考文献		356

算 法 概 述

算法是计算机学科中最具有方法性的核心概念，是计算机科学领域的基石之一，被誉为计算机学科的灵魂。

算法设计的优劣决定软件系统的性能，对算法进行研究能使我们深刻理解问题的本质及可能的求解技术。解决某个问题存在多种方法，寻求最优算法使得问题的解决更为方便和高效。所以我们不仅要为所解决的问题设计有效算法，还需要对算法进行分析，以追求算法性能的最优化。

本书涉及算法设计与算法分析两个阶段，算法设计的任务是为解决某一给定问题设计有效方法；算法分析的任务是在比较解决特定问题多种方法优劣的基础上寻求最优方法。算法设计与分析正是分析问题和解决问题的结合。

1.1 引 言

“算法”即演算法，中文名称出自《周髀算经》；而英文名称 Algorithm 是由 9 世纪波斯数学家提出的，他在数学上提出了算法这个概念。“算法”原为 algorism，意思是阿拉伯数字的运算法则，在 18 世纪演变为 algorithm。欧几里得算法被人们认为是史上第一个算法。第一次编写程序是 Ada Byron 于 1842 年为巴贝奇分析机编写求解伯努利方程的程序，因此 Ada Byron 被大多数人认为是世界上第一位程序员。查尔斯·巴贝奇(Charles Babbage)未能完成他的巴贝奇分析机，这个算法未能在巴贝奇分析机上执行。

一本早期的德文数学词典 *Vollständiges Mathematisches Lexicon* 《数学大全辞典》，给出了 algorithmus(算法)一词的如下定义：“在这个名称之下，组合了四种类型的算术计算的概念，即加法、乘法、减法、除法。”拉丁短语 Algorithmus Infinitesimalis(无限小方法)，在当时就用来表示 Leibnitz(莱布尼茨)所发明的以无限小量进行计算的微积分方法。

1950 年左右，algorithm 一词经常同欧几里得算法(Euclid's algorithm)联系在一起。欧几里得算法就是在欧几里得的《几何原本》(*Euclid's Elements*, 第七卷命题 i 和 ii)中所阐述的求两个数的最大公约数的过程(即辗转相除法)。

20 世纪,英国数学家图灵提出了著名的图灵论题,并提出一种假想的计算机抽象模型,这个模型被称为图灵机。图灵机的出现解决了算法定义的难题,图灵的思想对算法的发展起到了重要作用。

1.1.1 算法的描述

20 世纪 50 年代,欧几里得描述了求两个数的最大公约数的对程,被称为欧几里得算法。该算法又称辗转相除法,用于计算两个正整数 m 、 n 的最大公约数。其步骤如下:

步骤 1: 如果 $m < n$, 则交换 m 和 n 。

步骤 2: 令 r 是 m/n 的余数。

步骤 3: 如果 $r = 0$, 则输出 m ; 否则令 $m = n$, $n = r$ 并转向步骤 2。

其计算的原理依赖于下面的定理。

定理: $\gcd(m, n) = \gcd(n, m \bmod n)$ ($m > n$ 且 $m \bmod n$ 不为 0)。

计算两个数 m 、 n 的最大公约数,其输入为非负整数 m 和 n , 其中 m 、 n 不同时为零,输出为 m 与 n 的最大公约数。该算法对应的代码如下:

//迭代形式

```
int gcd(int m, int n) {
    if(m < n) gcd(n, m);
    int r;
    while(n != 0) {
        r = m % n;
        m = n;
        n = r;
    }
    return m;
}
```

//递归形式

```
int gcd(int m, int n) {
    if(m < n) gcd(n, m);
    if(n == 0)
        return m;
    else
        return gcd(n, m % n);
}
```

因此,在数学和计算机科学之中,算法(Algorithm)为一个计算的具体步骤,常用于计算、数据处理和自动推理。精确而言,算法是一个表示为有限长列表的有效方法。算法应包含清晰定义的指令用于计算函数。

1.1.2 算法的特性

算法设计的先驱者唐纳德·E.克努特(Donald E. Knuth)对算法的特征做了如下描述:

(1) 有穷性(Finiteness): 一个算法必须保证执行有限步之后结束,不能终止的过程不属于算法的范畴。

(2) 确切性(Definiteness): 算法的每一个步骤必须具有确切的定义, 即每一步要执行的动作是确切的, 是无二义性的。

(3) 可行性(Effectiveness): 一个算法是可行的, 就是算法中描述的操作都是可以通过已经实现的基本运算执行有限次来实现。并且, 在任何条件下, 算法只有唯一的一条执行路径, 即对于相同的输入只能得出相同的输出。

(4) 输入(Input): 一个算法必须具有 0 个或多个输入, 以刻画运算对象的初始情况。0 个输入适用于算法本身已确定了初始条件的情况。

(5) 输出(Output): 一个算法必须具有一个或多个输出, 以反映对输入数据加工后的结果。没有输出的算法是毫无意义的。

其中, 前 3 个特性较为集中地表现处理步骤, 后两个特性主要涉及输入/输出接口。

算法可用图 1.1 来描述。

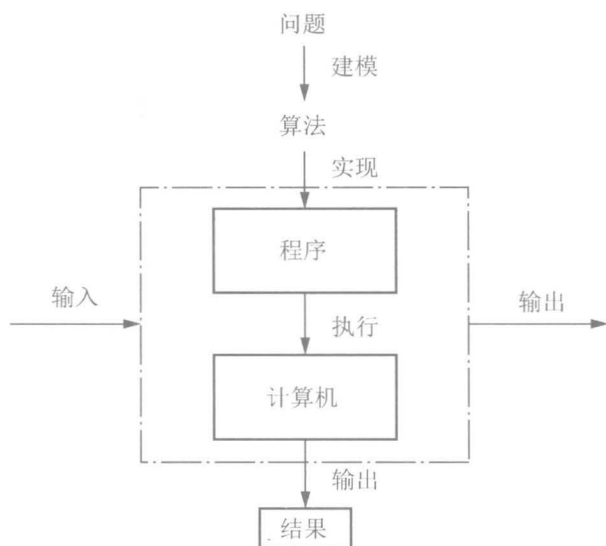


图 1.1 算法描述

1.1.3 为什么学习算法

1512 年 11 月 1 日, 在西斯廷教堂的天花板上, 人们看到了世界美术史上最大的壁画之一——《创世纪》。其中最引人瞩目的是上帝与亚当的形象。亚当被创造出来了: 一个有成熟的健美体格的美少年, 他那洋溢着青春的生命刚刚从睡幻中苏醒, 还不能站立起来行动, 等待着万能的造物主给他以力量。上帝耶和华被画成一位慈祥的老人, 扶着天使们飞来, 把有着无限力量的手伸向亚当, 而亚当将在握住上帝之手的刹那获得生命和力量, 作者以此表现出对解放人的力量的强烈渴望。《创世纪》(见图 1.2)的问世, 使当世最伟大的雕塑家米开朗基罗成为与达·芬奇并峙的最伟大的画家。这幅画里隐含着算法。

圣经上写着: 神 6 天创造天地万物, 第 7 日安歇。对于神创论者来说, 这是不可怀疑的事实, 但对于进化论者来说, 6 天创造一切根本就不可能。那圣经上为什么给出的是 6 天, 而不是其他的时间呢。