

信息与电子学科百本精品教材工程

新编计算机类本科规划教材

算法设计与分析

——C++语言描述

陈慧南 编著

<http://www.phei.com.cn>



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

新编计算机类本科规划教材

算法设计与分析

——C++语言描述

陈慧南 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书内容分为3部分：算法和算法分析、算法设计策略及求解困难问题。第1部分介绍问题求解方法、算法复杂度和分析、递归算法和递推关系；第2部分讨论常用的算法设计策略：基本搜索和遍历方法、分治法、贪心法、动态规划法、回溯法和分枝限界法；第3部分介绍NP完全问题、随机算法、近似算法和密码算法。书中还介绍了两种新的数据结构：跳表和伸展树，以及它们特定的算法分析方法，并对现代密码学做了简要论述。

本书结构清晰、内容翔实、逻辑严谨、深入浅出。书中算法有完整的C++程序，程序构思精巧，且有详细注释。所有程序都已在VC++环境下编译通过并能正确运行，它们既是学习算法设计的示例，也能使复杂抽象的算法设计更易于学习者理解和掌握。书中包含大量实例和图示，并附丰富的习题，便于自学。

本书可作为高等院校计算机科学与技术和其他相关专业的本科和研究生的“算法设计与分析”课程的教材或参考书，是“算法与数据结构”或“数据结构”课程有益的教学参考书，也可供计算机工作者和其他希望了解和学习算法知识的人员参考。

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。

图书在版编目(CIP)数据

算法设计与分析：C++语言描述/陈慧南编著. —北京：电子工业出版社，2006.5

新编计算机类本科规划教材

ISBN 7-121-02592-2

I. 算… II. 陈… 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 ③C语言—程序设计—高等学校—教材 IV. ①TP301.6 ②TP312

中国版本图书馆CIP数据核字(2006)第042904号

责任编辑：冉 哲

印 刷：北京牛山世兴印刷厂

出版发行：电子工业出版社

北京市海淀区万寿路173信箱 邮编 100036

经 销：各地新华书店

开 本：787×1092 1/16 印张：20.5 字数：524.8千字

印 次：2006年5月第1次印刷

印 数：5000册 定价：26.80元

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系。
联系电话：(010) 68279077。质量投诉请发邮件至 zllts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

前 言

算法设计与分析不仅是计算机科学与技术专业学生的必备知识，也是计算机应用工作者必不可少的基础知识。掌握扎实的算法设计与分析理论和方法有助于理工科学生进一步学习计算机技术，适应更广泛的职业挑战。

计算学科教学计划 2001 (Computing Curricula 2001, 简称 CC2001) 将计算机学科分成 14 个领域，每个领域分成若干个知识单元，每个知识单元又包括若干个主题。CC2001 强调算法，重视算法设计与分析能力和程序设计能力。计算机算法的基本内容主要包含在**算法与复杂性** (Algorithm And Complexity, 简称 AL) 和**程序设计基础** (Programming Fundamental, 简称 PL) 等知识领域中。在 CC2001 建议的计算机科学与技术专业的 280 个核心学时中，程序设计与算法方面分配 90 个核心学时，约占总核心学时的 32.1%。

算法领域涉及的内容广泛，通常包括迄今为止，算法学家们所设计的许多基本和经典算法，如排序、搜索、图算法、组合问题算法、字符串算法和大量的数值算法，算法问题求解、算法分析技术和常用的算法设计策略，可计算性理论和问题复杂性的研究，如计算模型、NP 完全问题和问题复杂度下界理论。近年来，算法研究在随机算法、近似算法、加密算法、分布式算法和并行算法，以及其他算法方面也都有很多新成果。

作为“算法设计与分析”课程教材，根据我国在算法与数据结构方面课程开设的实际情况，本书不再重复属于我国传统“数据结构”课程中的基本数据结构和算法的内容，但选用快速排序等在“数据结构”课程中已学过的若干排序、搜索和图算法，它们被作为算法设计策略和算法分析的实例使用。这种做法不是内容的简单重复，而是必要的和有益的深化。以学生熟知的知识为基础，介绍新知识，可使学生更容易理解和接受新的算法知识。

算法知识理论性较强，涉及的范围又很广，给学习和理解造成困难。为了将本书写成条理清晰、内容翔实、逻辑严谨、深入浅出的“算法设计与分析”教材，作者做了以下努力。

首先，本书分 3 部分组织内容，力求做到结构清晰、内容取舍恰当。

其次，书中算法都有完整的 C++ 程序，程序结构清楚，构思精巧，对程序代码都做了详细注释，所有程序都已在 VC++ 环境下编译通过并能正确运行，它们既是学习算法设计的示例，也是很好的 C++ 程序设计示例。

此外，本书通过大量实例和图示介绍算法，并有丰富的习题，便于自学。

这样做的目的是在保持算法科学性的同时，加强其技术性和实用性，也降低算法学习的难度，使复杂抽象的算法设计更容易为学习者理解和掌握。这也体现了计算机学科的科学性和工程性、理论性和实践性并重的学科特点。

全书包括 3 部分：算法和算法分析、算法设计策略及求解困难问题。

第 1 部分介绍算法概念、算法问题分类和问题求解方法，算法复杂度、递归技术，还介绍了两种新的数据结构：跳表和伸展树，以及它们特定的算法分析方法。

第 2 部分讨论常用的算法设计策略：基本搜索和遍历方法、分治法、贪心法、动态规划法、回溯法和分枝限界法。对于每种算法设计策略，通常先介绍一般方法，然后使用该策略解决若干经典的算法问题。

第3部分介绍 NP 完全问题、随机算法、近似算法，并对现代密码学和数论算法也做了简要论述。

本书作者在南京邮电大学讲授“算法设计与分析”和“数据结构”课程多年。本书是在作者编写出版的多本关于算法与数据结构领域教材的基础上，参考了近年来国内外多种算法设计与分析的优秀教材编写而成的。本书的编写得到了电子工业出版社的大力支持，并得到了南京邮电大学和计算机学院领导的推荐和关心，在此表示衷心感谢。

书中若有不妥之处，敬请读者批评指正。

作者

2006年3月

目 录

第 1 部分 算法和算法分析

第 1 章 算法问题求解基础	(3)
1.1 算法概述	(3)
1.1.1 什么是算法	(3)
1.1.2 为什么学习算法	(5)
1.2 问题求解方法	(5)
1.2.1 问题和问题求解	(6)
1.2.2 问题求解过程	(6)
1.2.3 系统生命周期	(7)
1.3 算法设计与分析	(7)
1.3.1 算法问题求解过程	(7)
1.3.2 如何设计算法	(8)
1.3.3 如何表示算法	(8)
1.3.4 如何确认算法	(9)
1.3.5 如何分析算法	(9)
1.4 递归和归纳	(9)
1.4.1 递归	(9)
1.4.2 递归算法示例	(11)
1.4.3 归纳证明	(14)
本章小结	(15)
习题 1	(15)
第 2 章 算法分析基础	(17)
2.1 算法复杂度	(17)
2.1.1 什么是好的算法	(17)
2.1.2 影响程序运行时间的因素	(18)
2.1.3 算法的时间复杂度	(19)
2.1.4 使用程序步分析算法	(20)
2.1.5 算法的空间复杂度	(21)
2.2 渐近表示法	(22)
2.2.1 大 O 记号	(22)
2.2.2 Ω 记号	(24)
2.2.3 Θ 记号	(24)
2.2.4 小 o 记号	(25)
2.2.5 算法按时间复杂度分类	(25)

2.3	递推关系	(26)
2.3.1	递推方程	(26)
2.3.2	替换方法	(27)
2.3.3	迭代方法	(27)
2.3.4	主方法	(28)
2.4	分摊分析	(29)
2.4.1	聚集方法	(30)
2.4.2	会计方法	(30)
2.4.3	势能方法	(31)
	本章小结	(32)
	习题 2	(32)
第 3 章	伸展树与跳表	(35)
3.1	伸展树	(35)
3.1.1	二叉搜索树	(35)
3.1.2	自调节树和伸展树	(36)
3.1.3	伸展操作	(36)
3.1.4	伸展树类	(38)
3.1.5	旋转的实现	(39)
3.1.6	插入运算的实现	(39)
3.1.7	分摊分析	(41)
3.2	跳表	(43)
3.2.1	什么是跳表	(43)
3.2.2	跳表类	(45)
3.2.3	级数分配	(46)
3.2.4	插入运算的实现	(47)
3.2.5	性能分析	(49)
	本章小结	(49)
	习题 3	(49)

第 2 部分 算法设计策略

第 4 章	基本搜索和遍历方法	(53)
4.1	基本概念	(53)
4.2	图的搜索和遍历	(54)
4.2.1	搜索方法	(54)
4.2.2	邻接表类	(55)
4.2.3	广度优先搜索	(56)
4.2.4	深度优先搜索	(58)
4.3	双连通分量	(61)
4.3.1	基本概念	(61)
4.3.2	发现关节点	(62)

4.3.3	构造双连通图	(66)
4.4	与或图	(66)
4.4.1	问题分解	(66)
4.4.2	判断与或树是否可解	(68)
4.4.3	构建解树	(69)
	本章小结	(71)
	习题 4	(71)
第 5 章	分治法	(73)
5.1	一般方法	(73)
5.1.1	分治法的基本思想	(73)
5.1.2	算法分析	(74)
5.1.3	数据结构	(75)
5.2	求最大最小元	(76)
5.2.1	分治法求解	(76)
5.2.2	时间分析	(77)
5.3	二分搜索	(78)
5.3.1	分治法求解	(78)
5.3.2	对半搜索	(79)
5.3.3	二叉判定树	(80)
5.3.4	搜索算法的时间下界	(82)
5.4	排序问题	(83)
5.4.1	合并排序	(83)
5.4.2	快速排序	(85)
5.4.3	排序算法的时间下界	(90)
5.5	选择问题	(91)
5.5.1	分治法求解	(92)
5.5.2	随机选择主元	(92)
5.5.3	线性时间选择算法	(94)
5.5.4	时间分析	(96)
5.5.5	允许重复元素的选择算法	(97)
5.6	斯特拉森矩阵乘法	(97)
5.6.1	分治法求解	(97)
5.6.2	斯特拉森分治法	(98)
	本章小结	(98)
	习题 5	(99)
第 6 章	贪心法	(102)
6.1	一般方法	(102)
6.2	背包问题	(103)
6.2.1	问题描述	(103)
6.2.2	贪心法求解	(104)

6.2.3	算法正确性	(105)
6.3	带时限的作业排序	(107)
6.3.1	问题描述	(107)
6.3.2	贪心法求解	(107)
6.3.3	算法正确性	(108)
6.3.4	可行性判定	(109)
6.3.5	作业排序贪心算法	(110)
6.3.6	一种改进算法	(111)
6.4	最佳合并模式	(114)
6.4.1	问题描述	(114)
6.4.2	贪心法求解	(115)
6.4.3	算法正确性	(116)
6.5	最小代价生成树	(117)
6.5.1	问题描述	(117)
6.5.2	贪心法求解	(118)
6.5.3	普里姆算法	(119)
6.5.4	克鲁斯卡尔算法	(121)
6.5.5	算法正确性	(123)
6.6	单源最短路径	(124)
6.6.1	问题描述	(124)
6.6.2	贪心法求解	(124)
6.6.3	迪杰斯特拉算法	(125)
6.6.4	算法正确性	(127)
6.7	磁带最优存储	(129)
6.7.1	单带最优存储	(129)
6.7.2	多带最优存储	(130)
6.8	贪心法的基本要素	(131)
6.8.1	最优量度标准	(131)
6.8.2	最优子结构	(132)
	本章小结	(132)
	习题 6	(132)
第 7 章	动态规划法	(135)
7.1	一般方法和基本要素	(135)
7.1.1	一般方法	(135)
7.1.2	基本要素	(136)
7.1.3	多段图问题	(136)
7.1.4	资源分配问题	(139)
7.1.5	关键路径问题	(140)
7.2	每对结点间的最短路径	(143)
7.2.1	问题描述	(143)

7.2.2	动态规划法求解	(143)
7.2.3	弗洛伊德算法	(144)
7.2.4	算法正确性	(145)
7.3	矩阵连乘	(146)
7.3.1	问题描述	(146)
7.3.2	动态规划法求解	(147)
7.3.3	矩阵连乘算法	(148)
7.3.4	备忘录方法	(150)
7.4	最长公共子序列	(151)
7.4.1	问题描述	(151)
7.4.2	动态规划法求解	(151)
7.4.3	最长公共子序列算法	(152)
7.4.4	算法的改进	(154)
7.5	最优二叉搜索树	(154)
7.5.1	问题描述	(154)
7.5.2	动态规划法求解	(155)
7.5.3	最优二叉搜索树算法	(157)
7.6	0/1 背包	(158)
7.6.1	问题描述	(158)
7.6.2	动态规划法求解	(159)
7.6.3	0/1 背包算法框架	(161)
7.6.4	0/1 背包算法	(164)
7.6.5	性能分析	(167)
7.6.6	使用启发式方法	(167)
7.7	流水作业调度	(168)
7.7.1	问题描述	(168)
7.7.2	动态规划法求解	(170)
7.7.3	Johnson 算法	(172)
	本章小结	(173)
	习题 7	(173)
第 8 章	回溯法	(175)
8.1	一般方法	(175)
8.1.1	基本概念	(175)
8.1.2	剪枝函数和回溯法	(176)
8.1.3	回溯法的效率分析	(178)
8.2	n -皇后	(179)
8.2.1	问题描述	(179)
8.2.2	回溯法求解	(179)
8.2.3	n -皇后算法	(180)
8.2.4	时间分析	(182)

8.3	子集和数	(183)
8.3.1	问题描述	(183)
8.3.2	回溯法求解	(183)
8.3.3	子集和数算法	(185)
8.4	图的着色	(186)
8.4.1	问题描述	(186)
8.4.2	回溯法求解	(187)
8.4.3	图着色算法	(187)
8.4.4	时间分析	(189)
8.5	哈密顿环	(189)
8.5.1	问题描述	(189)
8.5.2	哈密顿环算法	(189)
8.6	0/1 背包	(191)
8.6.1	问题描述	(191)
8.6.2	回溯法求解	(191)
8.6.3	限界函数	(192)
8.6.4	0/1 背包算法	(193)
8.7	批处理作业调度	(195)
8.7.1	问题描述	(195)
8.7.2	回溯法求解	(195)
8.7.3	批处理作业调度算法	(195)
	本章小结	(197)
	习题 8	(197)
第 9 章	分枝限界法	(199)
9.1	一般方法	(199)
9.1.1	分枝限界法概述	(199)
9.1.2	LC 分枝限界法	(201)
9.1.3	15 谜问题	(202)
9.2	求最优解的分枝限界法	(204)
9.2.1	上下界函数	(204)
9.2.2	FIFO 分枝限界法	(205)
9.2.3	LC 分枝限界法	(206)
9.3	带时限的作业排序	(207)
9.3.1	问题描述	(207)
9.3.2	分枝限界法求解	(207)
9.3.3	带时限作业排序算法	(208)
9.4	0/1 背包	(211)
9.4.1	问题描述	(211)
9.4.2	分枝限界法求解	(211)
9.4.3	0/1 背包算法	(212)

9.5 旅行商问题	(215)
9.5.1 问题描述	(215)
9.5.2 分枝限界法求解	(215)
9.6 批处理作业调度	(220)
9.6.1 问题描述	(220)
9.6.2 分枝限界法求解	(220)
9.6.3 批处理作业调度算法	(221)
本章小结	(224)
习题 9	(224)

第 3 部分 求解困难问题

第 10 章 NP 完全问题	(229)
10.1 基本概念	(229)
10.1.1 不确定算法和不确定机	(229)
10.1.2 可满足性问题	(232)
10.1.3 P 类和 NP 类问题	(233)
10.1.4 NP 难度和 NP 完全问题	(233)
10.2 Cook 定理和证明	(234)
10.2.1 Cook 定理	(234)
10.2.2 简化的不确定机模型	(234)
10.2.3 证明 Cook 定理	(235)
10.3 一些典型的 NP 完全问题	(240)
10.3.1 最大集团	(240)
10.3.2 顶点覆盖	(241)
10.3.3 3 元 CNF 可满足性	(242)
10.3.4 图的着色数	(243)
10.3.5 有向哈密顿环	(244)
10.3.6 恰切覆盖	(246)
10.3.7 子集和数	(248)
10.3.8 分划问题	(248)
本章小结	(249)
习题 10	(249)
第 11 章 随机算法	(251)
11.1 基本概念	(251)
11.1.1 随机算法概述	(251)
11.1.2 随机数发生器	(251)
11.1.3 随机算法分类	(251)
11.2 拉斯维加斯算法	(252)
11.2.1 标识重复元素算法	(252)
11.2.2 性能分析	(253)

11.3	蒙特卡罗算法	(254)
11.3.1	素数测试问题	(254)
11.3.2	伪素数测试	(254)
11.3.3	米勒-拉宾算法	(255)
11.3.4	性能分析	(256)
11.4	舍伍德算法	(257)
11.4.1	随机快速排序算法	(257)
11.4.2	舍伍德算法的其他应用	(257)
	本章小结	(258)
	习题 11	(258)
第 12 章	近似算法	(260)
12.1	近似算法的性能	(260)
12.1.1	基本概念	(260)
12.1.2	绝对性能保证	(260)
12.1.3	相对性能保证	(261)
12.1.4	近似方案	(262)
12.2	绝对近似算法	(262)
12.2.1	最多程序存储问题	(262)
12.2.2	NP 难度绝对近似算法	(263)
12.3	ϵ -近似算法	(264)
12.3.1	顶点覆盖近似算法	(264)
12.3.2	旅行商问题	(265)
12.3.3	NP 难度 ϵ -近似旅行商问题	(266)
12.3.4	具有三角不等式性质的旅行商问题	(267)
12.3.5	任务调度近似算法	(268)
12.4	$\epsilon(n)$ -近似算法	(271)
12.4.1	集合覆盖问题	(271)
12.4.2	集合覆盖近似算法	(272)
12.4.3	$\ln(n)$ -近似算法	(273)
12.5	多项式时间近似方案	(274)
12.5.1	任务调度近似方案	(274)
12.5.2	多项式时间近似方案	(275)
12.6	子集和数的完全多项式时间近似方案	(276)
12.6.1	子集和数指数时间算法	(276)
12.6.2	完全多项式时间近似方案	(277)
	本章小结	(279)
	习题 12	(279)
第 13 章	密码算法	(281)
13.1	信息安全和密码学	(281)
13.1.1	信息安全	(281)

13.1.2	什么是密码	(281)
13.1.3	密码体制	(282)
13.2	数论初步	(283)
13.3	背包密码算法	(285)
13.3.1	背包算法	(285)
13.3.2	超递增背包	(285)
13.3.3	由私人密钥产生公开密钥	(286)
13.3.4	加密方法	(287)
13.3.5	解密方法	(287)
13.3.6	背包安全性	(287)
13.4	RSA 算法	(288)
13.4.1	RSA 算法概述	(288)
13.4.2	RSA 的安全性	(289)
13.5	散列函数和消息认证	(289)
13.5.1	散列函数	(289)
13.5.2	散列函数结构	(290)
13.5.3	消息认证	(290)
13.6	数字签名	(291)
13.6.1	RSA 数字签名体制	(291)
13.6.2	需仲裁数字签名	(291)
	本章小结	(292)
	习题 13	(292)
附录 A	专有名词中英文对照表	(293)
附录 B	C++程序设计概要	(298)
B.1	函数与参数传递	(298)
B.2	动态存储分配	(301)
B.3	类与对象	(302)
B.4	函数和运算符重载	(303)
B.5	继承性和派生类	(304)
B.6	多态性、虚函数和动态联编	(305)
B.7	纯虚函数和抽象类	(306)
B.8	模板函数和模板类	(307)
B.9	友元函数和友元类	(310)
	参考文献	(313)

THE UNIVERSITY OF CHICAGO
DIVISION OF THE PHYSICAL SCIENCES
DEPARTMENT OF CHEMISTRY

第 1 章 算法问题求解基础

算法是计算机学科的一个重要分支，它是计算机科学的基础，更是计算机程序的基石。算法是计算机求解问题的特殊方法。学习算法，一方面需要学习求解计算领域中典型问题的各种有效算法，还要学习设计新算法和分析算法性能的方法。本章给出算法的基本概念，介绍使用计算机求解问题的过程和方法，讨论递归算法及证明递归算法正确性的归纳法。

1.1 算法概述

1.1.1 什么是算法

在学习一门计算机程序设计语言，如 C/C++ 或 Pascal 之后，应该对算法一词不再陌生。编写一个程序，实际上是在实现使用计算机求解某个问题的方法。在计算机科学中，算法一词用于描述一个可用计算机实现的问题求解 (problem-solving) 方法。

什么是算法？笼统地说，**算法 (algorithm)** 是求解一类问题的任意一种特殊的方法。较严格的说法是，一个算法是对特定问题求解步骤的一种描述，它是指令的有限序列。此外，算法具有下列 5 个特征。

- (1) **输入 (input)**: 算法有零个或多个输入量；
- (2) **输出 (output)**: 算法至少产生一个输出量；
- (3) **确定性 (definiteness)**: 算法的每一条指令都有确切的定义，没有二义性；
- (4) **能行性 (effectiveness)**: 算法的每一条指令必须足够基本，它们可以通过已经实现的基本运算执行有限次来实现；
- (5) **有穷性 (finiteness)**: 算法必须总能在执行有限步之后终止。

所有算法都必须具有以上 5 个特征。算法的输入是一个算法在开始前所需的最初的量，这些输入取自特定的值域。算法可以没有输入，但算法至少应产生一个输出，否则算法便失去了它存在的意义。算法是一个指令序列。一方面，每条指令的作用必须是明确、无歧义的。在算法中不允许出现诸如“计算 $5+3$ 或计算 $7-3$ ”这样的指令。另一方面，算法的每条指令必须是能行的。对一个计算机算法而言，能行性要求一条算法指令应当最终能够由执行有限条计算机指令来实现。例如，一般的整数算术运算是能行的，但如果 $1+3$ 的值需由无穷的十进制展开的实数表示，就不是能行的。因此，概括地说，算法是由一系列明确定义的基本指令序列所描述的，求解特定问题的过程。它能够对合法的输入，在有限时间内产生所要求的输出。如果取消有穷性限制，则只能称为**计算过程 (computational procedure)**。

描述一个算法有多种方法，可以用自然语言、流程图、伪代码和程序设计语言来描述。当一个算法使用计算机程序设计语言描述时，就是**程序 (program)**。算法必须可终止。计算机程序并没有这一限制，例如，一个操作系统是一个程序，却不是一个算法，一旦运行，只要计算机不关机，操作系统程序就不会终止运行。所以，操作系统程序是使用计算机语言描述的一个计算过程。

算法概念并不是计算机诞生以后才有的新概念。计算两个整数的最大公约数的辗转相除