

计算机算法与程序设计

朱 青 编著

清华大学出版社



计算机算法与程序设计

Computer Algorithm and Programming Design

朱 青 编著

清华大学出版社
北京

内 容 简 介

将本科“计算机算法与程序设计”课程与大学生程序设计竞赛有机地结合是新时期教学改革、培养实用型计算机优秀人才的创新。本书既系统深入地介绍算法设计的理论知识,又详尽地将其应用于实际编程,做到理论与实践的统一。

书中首先从理论的角度介绍了算法基础,数据抽象与数据结构,初等数论,组合数学初步;讲述了递归与分治策略,动态规划,贪心算法,搜索技术,图论算法;进一步研究了计算几何,排序算法;最后从实践的角度给出了程序设计典型实例及详细解析。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

计算机算法与程序设计/朱青编著. —北京:清华大学出版社,2009.10

ISBN 978-7-302-20267-7

I. 计… II. 朱… III. ①电子计算机—算法理论 ②程序设计 IV. TP301.6
TP311.1

中国版本图书馆 CIP 数据核字(2009)第 079272 号

责任编辑:汪汉友

责任校对:李建庄

责任印制:王秀菊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

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

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者:三河市春园印刷有限公司

经 销:全国新华书店

开 本:185×260 印 张:18.25

字 数:441 千字

版 次:2009 年 10 月第 1 版

印 次:2009 年 10 月第 1 次印刷

印 数:1~4000

定 价:27.00 元

本书如存在文字不清、漏印、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:010-62770177 转 3103 产品编号:032270-01

前 言

“计算机算法与程序设计”(computer algorithm and programming design)是计算机科学技术领域研究的重要基础课程,已成为众多理工科专业学生所喜爱的必修课之一。众所周知,计算机系统软件和应用软件,如操作系统、高级语言编译器、数据库管理系统及各种计算机应用软件,无论是单机环境的系统软件,还是分布式、并行计算软件都是程序设计的结晶,当然都离不开算法的设计与实现。计算机系统上的任何软件,都是按特定的算法进行设计并且编写必需的程序源代码予以实现的。算法性能的好坏,直接决定了所实现软件性能的优劣。对于算法的设计除了需要考虑算法自身的功能外,还需考虑算法的时空复杂度,关键是设计一个功能强、效率高、时空复杂性低的优化算法并用程序实现。因此,计算机算法与程序设计是计算机科学与技术的一个核心问题,也是大学计算机专业本科生的一门重要的专业基础课程。

本书的内容选材适当,循序渐进,互相衔接,逐步展开,具有系统性、先进性和实用性。

(1) 系统性。系统深入地介绍计算机专业基础课程“算法设计与分析”的理论知识;全面地讲解程序设计对于算法的设计与实现。全书包括:算法基础,数据抽象与数据结构,初等数论,组合数学初步;讲述了递归与分治策略,动态规划,贪心算法,搜索技术,图论算法;进一步研究了计算几何,排序算法;最后给出了程序设计典型实例。

(2) 先进性。计算机应用领域专家、优秀的重点高校教师、ACM-ICPC 国际大学生程序设计竞赛金牌教练,近年的科研与教学的积累与提炼,内容体系先进性与计算机算法、程序设计的有机结合的研究成果与经验结晶。

(3) 实用性。本书选材新颖,方法实用,题例丰富,取舍得当。采用 C++语言作为表述手段,书中提供了大部分算法的 C 程序和伪码算法,尽量使算法的描述简明、清晰,做到从算法到程序设计逐步求精。为了加深对知识的理解,书的最后加入一章难易适当的例题和经典习题,以适应不同程度读者学习需要。

教材特色体现在,将复杂的算法通过 C++语言程序设计实现,做到难易的有机结合,尤其是将抽象数据结构 ADT 有机的融于算法教材,书的最后有大量的程序设计典型实例、详细的设计方法与算法分析,并用程序实现,使读者既系统深入地掌握理论知识,又能将其应用于实际编程,做到理论与实践的统一。

随着信息技术的发展,计算机算法与程序设计的普及,依据其难易等级,已从大学本科课程、研究生基础教学扩展到中学、高职高专教育;尤其是计算机信息类 ACM(Association for Computing Machinery)国际大学生程序设计竞赛(ACM International Collegiate Programming Contest, ACM-ICPC),中学信息学奥林匹克国际竞赛,已从计算机算法与程序设计研究的高端到低端全面展开。其目的是培养学生良好的程序设计技巧和熟练的算法分析能力,能够开发出高效率的有效高级语言程序。同时,“计算机算法与程序设计”的本科课程目前在各个高校普遍开设,是一门计算机学术研究的专业性基础课程,尤其是将

大学生程序设计竞赛与本科“算法与程序设计”课程有机地结合起来是新时期教学改革、培养创新型、国际化计算机优秀人才的有价值、有意义的尝试。

顺应时代发展的需求，计算机算法与程序设计技术，计算机等级考试，已经像掌握英语、驾驶汽车技术一样成为每一个新时代公民的必备基础。它在计算机知识结构中的地位与作用是不可低估的。事实说明，要想有效地使用计算机，仅掌握计算机语言而缺乏数据结构和算法的有关知识，难以应付众多复杂的应用课题。为此，计算机算法与程序设计有其重要的理论意义与实际应用需求。

感谢清华大学出版社的大力支持；感谢丁博麟同志为本书所做的工作；感谢中国人民大学 ACM-ICPC 国际大学生程序设计代表队的教练、老师和队员们为本书所做的工作；在本书编写过程中还得到其他同志的帮助，在此一并表示诚挚的感谢。

由于水平有限，书中不当之处，敬请读者指正。

朱 青 2009 年 8 月
中国人民大学

目 录

第 1 章 绪论	1
1.1 算法研究的意义	1
1.2 算法与程序	7
1.3 算法的描述工具	13
1.4 算法的复杂性分析	16
1.4.1 时间复杂度	16
1.4.2 空间复杂度	19
1.5 常用数学分析公式	21
第 2 章 数据抽象与数据结构	24
2.1 数据抽象概念	24
2.1.1 数据结构的基本概念和术语	24
2.1.2 数据抽象	25
2.2 基本数据结构	28
2.2.1 线性表与向量	28
2.2.2 链表	31
2.2.3 栈和队列	33
2.2.4 二叉树	39
2.2.5 图	42
2.3 关键数据结构拓广	45
2.3.1 哈希表	45
2.3.2 并查集(等价类)	48
2.3.3 线段树	50
2.3.4 二叉堆	52
第 3 章 初等数论	55
3.1 数论基础	55
3.1.1 素数与算术基本定理	55
3.1.2 最大公约数与最小公倍数	56
3.2 同余方程	58
3.2.1 同余方程概念	58
3.2.2 中国剩余定理	60
3.3 数论函数	61
3.3.1 欧拉函数	61

3.3.2	积性函数	63
3.4	素数和整除	64
3.4.1	筛法求素数	64
3.4.2	整数 N 的因子函数	65
3.5	高精度计算	67
第 4 章	组合数学初步	72
4.1	加法原理与乘法原理	72
4.2	鸽笼原理和 Ramsey 数	74
4.3	递推关系和生成函数	76
4.3.1	Fibonacci 数	77
4.3.2	Catalan 数	78
4.3.3	第二类 Stirling 数	81
4.4	排列组合	82
4.4.1	字典序排列	82
4.4.2	组合算法	83
4.4.3	二项式系数	85
4.5	容斥原理	86
4.5.1	容斥原理的概念	86
4.5.2	错排问题	87
4.6	Polya 定理及其应用	89
第 5 章	递归与分治策略	94
5.1	递归概念	94
5.1.1	递归与递归调用	94
5.1.2	递归应用	98
5.2	分治法概述	100
5.2.1	分治法基本思想	100
5.2.2	分治算法设计和特点	101
5.3	分治法的基本应用	103
5.3.1	最大最小值	103
5.3.2	Strassen 矩阵乘法	104
5.4	分治法解骑士周游	107
5.5	大整数乘法	109
5.5.1	常规大整数乘法	109
5.5.2	分治法解大整数乘法	111
5.6	棋盘覆盖问题	112

第 6 章	贪心算法	114
6.1	贪心算法概述	114
6.1.1	贪心举例	114
6.1.2	贪心算法的理论基础	116
6.1.3	贪心算法与动态规划算法的区别	117
6.2	背包问题	118
6.3	机器任务调度算法	119
6.3.1	多机调度问题	119
6.3.2	活动安排问题	120
6.4	最小生成树	122
6.4.1	普里姆 (Prim) 算法	122
6.4.2	克鲁斯卡尔 (Kruskal) 算法	123
6.5	哈夫曼 (Huffman) 树及其应用	126
6.5.1	Huffman 树	126
6.5.2	哈夫曼编码	129
6.5.3	Huffman 算法的正确性	130
第 7 章	动态规划	132
7.1	动态规划算法思想	132
7.1.1	动态规划最优决策原理	132
7.1.2	动态规划求解步骤	134
7.1.3	动态规划的数学抽象	137
7.2	矩阵连乘问题	138
7.3	最长子序列探索	141
7.3.1	最长递增子序列	141
7.3.2	最长公共子序列	143
7.4	多段图的最短路径	145
7.5	资源分配问题	149
7.6	树状动态规划	151
第 8 章	搜索技术	154
8.1	盲目搜索算法	154
8.1.1	对分搜索	154
8.1.2	DFS 与 BFS 搜索算法	156
8.1.3	盲目搜索算法应用	160
8.2	回溯算法	162
8.3	启发式搜索	165
8.3.1	启发式搜索策略	165
8.3.2	A^* 算法	167

8.4	博弈问题	170
8.4.1	博弈树	170
8.4.2	极小极大搜索法	172
8.5	α - β 剪枝技术	173
第 9 章	图论算法	176
9.1	基本概念和定理	176
9.1.1	可行遍性问题	177
9.1.2	平面图	177
9.1.3	独立集、覆盖与支配集	180
9.2	最短路径	182
9.2.1	Dijkstra 算法	182
9.2.2	Floyd 算法求一对点最短路径	184
9.3	道路和回路	186
9.3.1	欧拉道路和欧拉回路	186
9.3.2	哈密尔顿图和货郎担问题	188
9.4	网络流算法	193
9.4.1	基本概念	193
9.4.2	最大流问题	195
9.4.3	最小费用流	198
9.5	二分图相关问题	200
9.5.1	二分图的最大匹配	200
9.5.2	二分图的最佳匹配	203
第 10 章	计算几何	207
10.1	计算几何基本问题	207
10.1.1	矢量与线段	207
10.1.2	几何计算公式	209
10.2	点与线段的关系	212
10.2.1	点与线段的距离	212
10.2.2	线段与直线的交点	214
10.3	多边形	217
10.3.1	多边形基本概念	217
10.3.2	点与多边形的关系	218
10.4	凸包问题	219
10.4.1	判断凸包	219
10.4.2	寻找凸包	220
10.5	欧拉定理及其应用	221

第 11 章	排序	224
11.1	排序基础	224
11.2	比较排序法	225
11.2.1	插入排序	225
11.2.2	冒泡排序	227
11.2.3	简单选择排序	228
11.3	基于分治策略的排序算法	229
11.3.1	快速排序	229
11.3.2	归并排序	233
11.4	堆排序	235
11.4.1	树状选择排序	235
11.4.2	堆排序	235
11.5	基数排序	238
11.6	排序小结	241
第 12 章	算法与程序经典实例	244
12.1	计算机算法设计实例	244
12.2	国际竞赛程序实例分析	252

第1章 绪 论

计算机算法与程序设计 (Computer Algorithm and Programming Design) 是计算机科学技术领域研究的重要基础课程, 目前在各个高校普遍开设的本科课程, 已成为众多理工科专业学生所喜爱的选修课之一。算法 (algorithm) 是一组有限规则, 即为某个特定问题提供了计算机求解的运算序列。通俗点说, 就是计算机解题的过程。算法分为并行算法和“传统意义”上的单处理器计算机上执行的算法, 本书重点研究后者, 重点讲述构成算法与程序的基本方法, 解题思路, 求解过程, 求解效果的优劣分析等重要特征。

本章主要内容, 1.1 节算法研究的意义, 列举多个实例, 详细讨论“算法”的概念和研究算法的意义; 1.2 节算法与程序, 讲述算法如何逐步求精, 实现程序设计; 1.3 节算法的描述工具, 讨论算法的伪代码表示, 说明算法的精确描述工具; 1.4 节算法的复杂性分析, 简要介绍算法分析技术, 研究时间复杂度与空间复杂度分析; 1.5 节常用数学分析公式, 提出在算法分析中实施计算的一些必备数学基础, 这些方法将会帮助设计和分析算法。

1.1 算法研究的意义

随着信息技术的发展, 计算机算法与程序设计的普及, 依据其难易等级, 已从大学本科课程、研究生基础教学扩展到中学、高职高专教育; 尤其是计算机信息类 ACM 国际大学生程序设计竞赛 (ACM International Collegiate Programming Contest, ACM-ICPC), 中学信息学奥林匹克国际竞赛, 已从计算机算法与程序设计研究的高端到低端全面展开。其目的是培养学生良好的程序设计技巧和熟练的算法分析能力, 能够开发出高效率的有效高级语言程序。

顺应时代发展的需求, 掌握计算机算法与程序设计技术已经像掌握英语、驾驶汽车技术一样, 成为每一个新时代公民的必备基础。事实说明, 要想有效地使用计算机, 仅掌握计算机语言而缺乏数据结构和算法的有关知识, 难以应付众多复杂的应用课题。为此, 首先简单回顾一下算法与程序设计学科形成和发展的历史。

在电子计算机发展的初期, 人们使用计算机主要是处理数值性问题, 解决人们用手工或机械计算机难于胜任的数值计算。当时所涉及的操作对象还都比较简单, 不外乎是整型、实型和布尔型数据; 所编写的程序并不复杂。以此为对象的程序设计可称为数值型程序设计, 对应的软件称为“数学软件”。

随着电子计算机使用领域的扩大和深入, 解决“非数值性问题”越来越引起人们的关注, 例如金融和工商企业管理信息系统, 支持多媒体的文献资料查询、神经元和模式识别, 网络与通信, 图形化用户界面技术等。解决此类问题使用的数学工具已不是分析数学及其计算方法, 而更多地用到离散数学和计算机的有关知识, 所涉及的数据也更复杂, 其突出的特点是, 数据元素之间所具有的特定联系已不能用分析数学的方程式来简单描述。数据包括数值性、非数值性, 对数据的处理的复杂性大幅度提高, 直接编写大规模复杂程

序将面临许多困难，由此必须借助于计算机算法设计指导程序设计的实现。

为了帮助读者建立对算法与程序设计的感性认识，下面先举几个简单应用实例。

例 1.1 求解 Fibonacci 数列（斐波那契序列）前 40 个数问题。

Fibonacci 数列的特点：第 1, 2 个数分别是 1, 1。从第 3 个数开始，该数是其前面两个数之和。Fib(*n*)表示 Fibonacci 数列的第 *n* 项，*n* 从 1 开始。由此，Fibonacci 数列可以由如下数学表达式定义给出：

$$\begin{aligned} \text{Fib}(1) &= 1; \\ \text{Fib}(2) &= 1; \\ &\vdots \\ \text{Fib}(n) &= \text{Fib}(n-1) + \text{Fib}(n-2); \quad (\text{当 } n > 2 \text{ 时}) \end{aligned}$$

这是一个有趣的古典数学问题：有一对兔子，从出生后第 3 个月起每个月都生一对兔子。小兔子长到第 3 个月后每个月又生一对兔子。假设所有兔子都不死，问每个月的兔子总数为多少？分析这个问题，不满 1 个月的为小兔子，满 1 个月不满 2 个月的为中兔子，满 3 个月以上的为老兔子，老兔子每个月可以繁殖 1 对小兔子。如图 1.1 所示，可以求得每个月的兔子总数依次为 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, …。这就是 Fibonacci 数列。

如果用计算机来求解问题的话，一个直接的有公式自然解法导致 C 语言程序如下：

```
long fib(int n)
{ if (n == 1 || n == 2) return 1;
  else return fib(n-1)+fib(n-2);
}
```

可以看见，代码描述简单，此方法是递归算法，以直接调用自身函数 fib(*n*)来计算 Fibonacci 数列，就引起这个函数 fib(*n*)的递归调用活动。它的频繁程度如何呢？我们注意到，对于 *n* > 2，每调用一次引起两个新的调用，因此，调用的总次数按指数增长（如图 1.2 所示）。这样一个程序其运算效率是很低的。

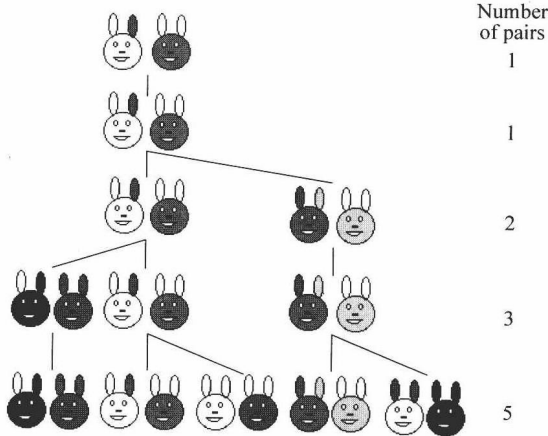


图 1.1 五对兔子的家族树示意图

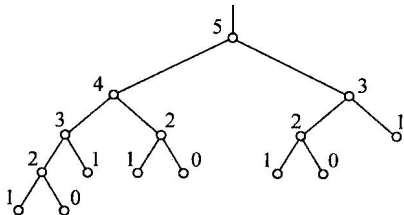


图 1.2 fib(5)的 15 次调用示意图

如果利用两个辅助变量 *f*₁, *f*₂ 分别代替 fib(*n*)和 fib(*n*-1)，而采用迭代算法来计算斐波那

契序列，可以避免同一值的重复调用计算从而提高效率。其非递归迭代算法描述如下：

```

long int fib(n)
{
    long int f1,f2;
    int i,
    f1=1; f2=1;
    for (i=1;i<=20;i++)
    {
        f1=f1+f2;
        f2=f2+f1;
    }
}

```

因此，求解 Fibonacci 数列有递归和非递归两种算法。对于递归算法易于理解并很好编写程序实现，但是效率低。研究实验表明：每一个递归程序都可以翻译成等价的迭代形式的程序。但这包含着对递归栈的显式处理，而这些运算常常模糊了程序的本质，以致使它难以理解。如果本质上是递归的而不是迭代的算法应当表述成递归过程。另一方面要考虑程序设计的目标与要求。依此决定算法的选用。求解 Fibonacci 数列抽象的递归算法模型是一类具有代表性的计算机算法。需要注意，按照要求添加适当的输入/输出语句。在许多“数值型”计算与“非数值型”计算中经常会用到，如果一个问题可以转化为一个结构相同，程序规模更小的问题，可以选用递归算法。

例 1.2 生活中的共进晚餐座位安排。

在一次学术会议上，有 11 位学者打算几天都在同一个圆桌上共进晚餐，并且希望每次晚餐时，每位学者两边邻座的人都不相同。按照这样一种要求，他们在一起共进晚餐最多能有几天？

分析这个问题可以抽象为： N 个顶点的完备图共有 $N(N-1)/2$ 条边的图论问题，每一条汉密尔顿环有 N 条边，因为要求任意两条汉密尔顿环无公共边，所以至多有 $(N-1)/2$ 条汉密尔顿环。现在 $N=11$ ，所以至多有 5 条两两无公共边的汉密尔顿环。

图 1.3 (a) 所示，存在一条汉密尔顿环 (1, 2, 3, ..., 11)，将此图的顶点标号分别旋转： $360/10$ 度得到图 1.3 (b)； $2 \times 360/10$ 度得到图 1.3 (c)； $3 \times 360/10$ 度得到图 1.3 (d)； $4 \times 360/10$ 度得到图 1.3 (e)。每个图对应一个汉密尔顿环。

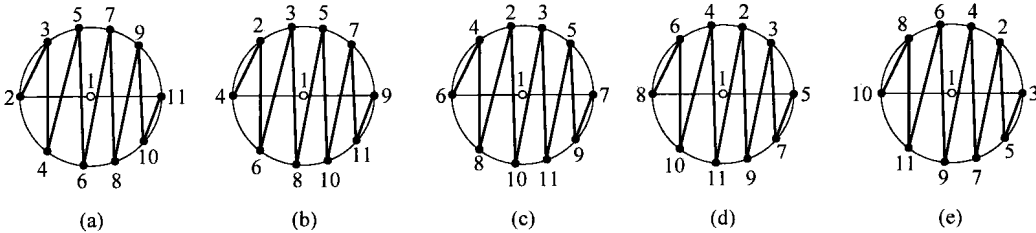


图 1.3 共进晚餐座位安排示意图

因此，按照题中要求能共进晚餐 5 天，如果这 11 个学者分别标记为 1, 2, ..., 11，那么 5 天中每天 11 个人的环形排列如下：

- (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11)
- (1, 4, 2, 6, 3, 8, 5, 10, 7, 11, 9)
- (1, 6, 4, 8, 2, 10, 3, 11, 5, 9, 7)
- (1, 8, 6, 10, 4, 11, 2, 9, 3, 7, 5)

(1, 10, 8, 11, 6, 9, 4, 7, 2, 5, 3)

求解此生活中的共进晚餐座位安排问题抽象为经典的汉密尔顿环算法模型，这是图论算法研究中具有代表性的计算机算法。

例 1.3 平面地图染色问题。

对平面地图染色问题，就是使地图边界相邻的国家有不同的颜色，如图 1.4 所示。平面地图抽象为一种平面图，它的“面”代表国家，“边”表示国家之间的边界，而“点”则是边界的交汇处。假设地图里所有区域都是连通的，平面地图 Graph 染色问题，显然，对地图 Graph 的面染色问题等价于对 Graph 的对偶图 Graph 的顶点进行染色的问题。求平面图的色数等于是求平面地图染色所需要的最少颜色数，使得没有两个相邻的区域指定为相同的颜色。

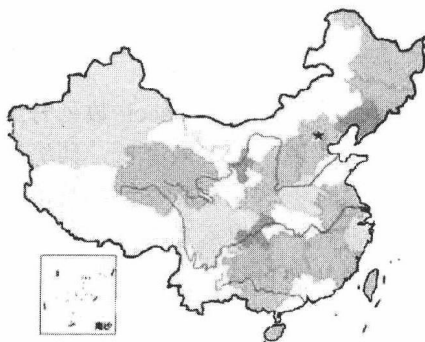


图 1.4 平面地图染色

考虑图 G 的着色问题可以表示为：给定一个无向图 $G(V, E)$ ，使用四种颜色之一（四种颜色分别是 1, 2, 3 和 4）为 V 中的每个顶点染色，使得没有两个邻接的顶点有同样的颜色，称这样的染色是合法的；如果两个邻接顶点着同一种颜色则是非法的。一种染色可以用 N 元组 $(C_1, C_2, \dots, C_n)$ 来表示，使 $C_i \in \{1, 2, 3, 4\}$, $1 \leq i \leq N$ 。例如， $(1, 2, 2, 3, 1, 4)$ 表示 6 个顶点的图的着色。一个 N 个顶点的图共有 4^N 种可能的着色（合法的和非法的）。问题是给出一个图最少需要染几种颜色。图的染色问题在历史上是有名的难题，人们做过大量的研究，因为它“难”而又有用，因此，引起人们的广泛关注。由图的染色问题引申出许多有趣的难题是对人类智慧的挑战。

四染色问题（four-color problem）连通简单平面图的色数不超过 4。1852 年，盖思里（Guthrie）把四色猜想转告了他的老师德·摩根，德·摩根对这个问题极其感兴趣，并且向数学界公布了四色猜想。四色猜想曾一度被列为与数论中的 Fermat 猜想，函数论中的 Riemann 假设相提并论的三大数学难题之一，受到世界上许多最有才华的数学家的研究。历史上曾有许多人宣布证明了它，但都被后人否定。1976 年，美国数学家肯尼思·阿佩尔（Kenneth Appel）和沃尔夫冈·黑肯（Wolfgang Haken）宣布了一个借助于计算机的证明。但是，这个证明引起了争论。原因在于计算复杂度太高。常见的依次染色算法：即图的染色问题的启发式算法，依据“四染色”定理，即可以用不多于四种颜色对地图染色，使相邻的行政区域不重色。

应用“四染色”定理的结论，用回溯（Backtracking）算法对一幅给定的地图染色。算法思想：从第一号行政区开始逐一染色，每一个区域逐次用颜色 1、2、3、4 进行试探。若当前所取的色数与周围已染色的行政区不重色，则用栈记下该行政区的色数，否则依次用下一色数进行试探；若出现用 1~4 色均与相邻区域发生重色，则需退栈回溯，修改当前栈顶的色数，再进行试探。直至所有行政区域都已分配合适的颜色。但是算法所产生的染色解离最优解相差很远。求解四染色问题抽象为“回溯”算法模型，是一类具有代表性的计算机算法。回溯算法把要求解的问题当做一系列的决策来考虑。对每一个决策试探地分析可能出现的结果。在这个意义上，回溯算法与穷举算法相类似。但是，二者的解空间不同，

回溯算法可以由试探过程自己修整，因此有较好的执行效率，在计算机科学与技术中应用广泛。

例 1.4 求解 N 个城市之间敷设通信光缆的最低经济代价的最小费用问题。

在 N 个城市之间敷设通信光缆，每两个城市之间的敷设通信光缆费用可以估算出来，施工设计部门将各城市之间的通信光缆的造价用一个代价图，如图 1.5 (a) 表示，结点代表各个城市，边及附带的数据表示一对城市之间敷设通信光缆的估算费用。问题是求解敷设通信光缆的最低经济代价最小费用问题。这种抽象的数据结构表示是带权图结构，可以使用相应的数组或者邻接表来描述图和存储图，通过求图的最小生成树算法找到符合要求的敷设方案，如图 1.5(b)所示，按照这种方案施工其敷设通信光缆的费用最少。

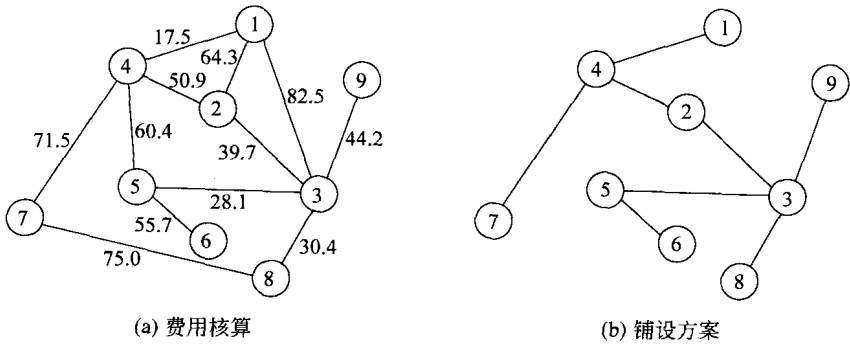


图 1.5 敷设通信光缆示意图

网的最小生成树问题可以利用普里姆 (Prim) 算法或者克鲁斯卡尔算法求网的最小生成树。若要在 n 个城市间敷设通信光缆，只需要敷设 $n-1$ 条通信光缆即可。重点考虑如何以最低的经济代价建设这个通信网。对于算法的选用原则：

- (1) 普里姆 (Prim) 算法：适合于求边稠密带权图的最小生成树，时间复杂度为 $O(n^2)$ 。
- (2) 克鲁斯卡尔 (Kruskal) 算法：适合于求稀疏带权图的网的最小生成树，时间复杂度为 $O(e \log e)$ (e 为网中边数)。

这里重点讨论普里姆算法求解最小生成树，以解决 N 个城市之间敷设通信光缆的最低经济代价的最小费用问题。

最小生成树的办法解决即普里姆 (Prim) 算法：把 N 个城市看作 N 个顶点，把可以敷设通信光缆的两个城市间用边连起来，把相连的两个城市之间的敷设造价费用作为这一条边的权值。将所有边的权值按照大小排列起来，先选择权最小的边的两个顶点并入集合 S ，再选择第二条边，把与这条边关联的顶点并入集合 S ，其约束条件是，所选择的边不能与集合 S 中所选择的边构成回路，如果有权相同的边时则任选一条。按此约束条件依次选取……，当所选的边的个数为 $N-1$ 时就停止运算。此时选择的线路即为敷设通信光缆的最优线路。

形式化描述：

假设 $\text{Network}=(V, E)$ 是一个带权图，最小生成树 $\text{Tree}=(V, \text{Tree_E})$

其中， Tree_E 是 N 上最小生成树中边的集合；

U 是 Network 上最小生成树中点的集合，并且 U 是 V 的子集；

算法初始化从 $U=\{u_0 | (u_0 \in V)\}$ ， $\text{Tree_E}=\{\}$ 开始。

算法流程:

在所有 $u \in U, v \in V-U$ 的边 $(u, v) \in E$ 中找一条权值最小的边 (u, v) 。

并入集合 Tree_E , 同时 v 并入 U 。

重复执行上述操作, 直至最小生成树中点的集合 $U=V$ 为止。

此时最小生成树中边的集合 Tree_E 一定有 $n-1$ 边 (这是有定理保证的), 则树 $\text{Tree}=(V, \text{Tree_E})$ 为带权图 Network 的最小生成树。

另外, 为实现这个算法需附设一个辅助数组 Cost_edge , 记录从 U 到 $V-U$ 具有最小代价的边 $\text{cost}(u, v)$ 表示边 (u, v) 的权。对每个顶点 $v_i \in V-U$, 在辅助数组中存在一个相应分量 $\text{Cost_edge}[i-1]$, 它包括两个域, 其中:

low 存储该边上的权, $\text{cost_edge}[i-1].\text{low} = \text{Min}\{\text{cost}(u, v_i) | v_i \in V-U, u \in U\}$;

vex 域存储该边依附的在 U 中的顶点, $\text{cost_edge}[i-1].\text{vex} = \{v_i | v_i \in V-U\}$ 。

对于连通的带权图, 其生成树也是带权生成树, 各边的权值总和称为该树的权, 权最小的生成树称为最小生成树 (minimum spanning tree, MST)。

为求解 N 个城市之间敷设通信光缆的最低经济代价的最小费用问题, 选用解决带权图的最小生成树普里姆 (Prim) 算法, 这是计算机算法与程序设计的经典算法, 由此可见算法在解决数据结构中图、带权图、树、最小生成树等的应用中所起的重要作用。

例 1.5 无线自组织网络进行通信所必备的基本操作: 组播与广播技术。

有线与无线网络的传播问题可以抽象为树结构、图结构及其最小生成树的算法。例如, 无线网络中的广播与组播是两种散布传输数据的基本操作。组播操作是将数据包从一个结点 (即源结点) 传输到网络中的一个结点集 (即一组给定的结点)。广播操作是组播操作的一种特殊情况, 是将数据包从源结点传输网络中其他所有结点的操作。在这种多跳并且电能有限的网络中, 广播和组播操作发挥着非常重要的作用, 许多路由算法需要用这两种操作去更新维护结点间的路由信息及状态。广播和组播操作还被广泛地用来在网络中散布和收集监测数据。因此, 设计能量有效的 (即节省能量的) 广播和组播协议具有重要价值。

设计能量有效的组播 (广播) 路由协议的首要问题是, 如何度量协议的能量有效性? 目前主要的度量方式之一: 在一次组播 (广播) 操作中, 所有结点能量消耗的和的最小化。由于最小化能量消耗的组播 (广播) 问题属于 NPC, 为了解决这类困难的组合优化问题, 一种可行的方法是建立线性规划模型。为这个问题建立了混合整数线性规划的模型, 在结点数量较少的情况下可以利用分支定界法或割平面法可解出最优解。

全向天线模型下的最小化能量广播算法: 一个直接的方法是利用最小支撑树和最短路径树去近似最小能量广播树。最早提出的著名 BIP 算法与标准 Prim 算法原理相同。BIP 算法维护了一个从源结点发出的树, 利用无线通信特性一次一个往树中添加结点。与 Prim 算法不同的是, 新结点一次一个的根据最小增加能量添加到树中, 而不是最小权值。算法的时间复杂度为 $O(n^3)$ 。由此可见, 计算机算法及其改进算法对新领域新问题的解决起到至关重要的作用。

例 1.6 无线网络中, 传感器定位技术的计算。

无线传感器网络由于其潜在的商业应用和军事应用价值受到了广泛的关注。最新的研究表明无线传感器网络可以在人们生产生活的多个方面发挥作用, 例如, 环境监测、交通流量控制、农业病虫害监控、森林防火甚至是军事领域, 无线传感器网络都发挥出传统网

络所无法达到的诸多作用。

在传感器网络中，位置信息对传感器网络的监测活动至关重要，事件发生的位置或获取信息的结点位置是传感器结点监测消息中所包含的重要信息。如在环境监测应用中，需要知道采集环境的具体区域位置，如图 1.6 所示；对于突发事件，如需知道森林火灾位置，战场敌方车辆位置，天然气管道泄漏的具体地点等。因此传感器结点必须能够在布放后实时地进行定位。即定位是根据少数已知位置的结点，确定传感器结点监测到的事件发生的具体位置。可见定位技术的重要性。

无线网络中，通信结点发射信号可以覆盖的区域通常用一个圆心在这个结点的圆表示。当这个结点发射通信信号时，所有在这个区域内的结点均可以收到这个信号。这里研究的问题是定位技术：可以选择三边测量法计算结点位置公式。

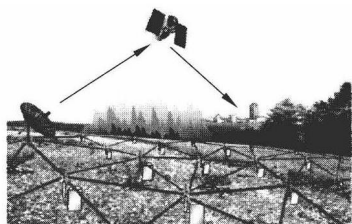


图 1.6 传感器森林防火检测

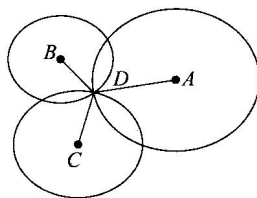


图 1.7 三边定位测量法图示

三边定位测量法，如图 1.7 所示，已知 A 、 B 、 C 三个结点的坐标分别为 (x_1, y_1) 、 (x_2, y_2) 、 (x_3, y_3) ， A 、 B 、 C 到未知结点 D 的距离分别为 d_1 ， d_2 ， d_3 ，假设结点 D 的坐标设定为 (x, y) 。那么，存在下列公式：

$$\begin{aligned} d_1 &= \sqrt{(x - x_1)^2 + (y - y_1)^2} \\ d_2 &= \sqrt{(x - x_2)^2 + (y - y_2)^2} \\ d_3 &= \sqrt{(x - x_3)^2 + (y - y_3)^2} \end{aligned}$$

得到 D 点的坐标 (x, y) ：

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 2(x_1 - x_3) & 2(y_1 - y_3) \\ 2(x_2 - x_3) & 2(y_2 - y_3) \end{bmatrix} + \begin{bmatrix} x_1^2 - x_3^2 + y_1^2 - y_3^2 + d_3^2 - d_1^2 \\ x_2^2 - x_3^2 + y_2^2 - y_3^2 + d_3^2 - d_2^2 \end{bmatrix}$$

可见计算几何是计算机算法与程序设计的重要组成部分。

总之，定位信息除用来报告事件发生的地点，还具有目标跟踪，实时监视目标行动路线，预测目标前进轨迹的功能；选择网络路由，避免信息在整个网络中的扩散，并可以实现定向的信息查询进行网络管理。

1.2 算法与程序

“算法”通常是指能够被计算机执行的一种解法。也就是在有限时间内计算机执行的一条指令方法。在这个过程中，无论是形成解题思路还是编写程序，都是在实施某种算法。前者是推理实现的算法，后者是操作实现的算法。计算机解题的核心是算法设计与程序设计的实现。一个算法应该具有以下 5 个重要特征。

(1) 输入：算法所接受的输入数据集。