

算法设计与分析

SUANFA SHEJI
YU
FENXI

项丽萍 著

算法设计与分析

ALGORITHM DESIGN
AND ANALYSIS

第2版

算法设计与分析

SUANFA SHEJI
YU
FENXI

项丽萍 著

图书在版编目 (C I P) 数据

算法设计与分析 / 项丽萍著. — 太原: 山西人民出版社, 2009.4

ISBN 978-7-203-06375-9

I. 算… II. 项… III. ①电子计算机—算法设计②电子计算机—算法分析 IV. TP301.6

中国版本图书馆CIP数据核字(2009)第017330号

算法设计与分析

著 者: 项丽萍

责任编辑: 冯灵芝

装帧设计: 清晨阳光 (谢成) 工作室

出 版 者: 山西出版集团·山西人民出版社

地 址: 太原市建设南路 21 号

邮 编: 030012

发行营销: 0351-4922220 4955996 4956039

0351-4922127 (传真) 4956038 (邮购)

E-mail: sxskcb@163.com 发行部

sxskcb@126.com 总编室

网 址: www.sxskcb.com

经 销 者: 山西出版集团·山西人民出版社

承 印 者: 山西嘉祥印刷包装有限公司

开 本: 850mm × 1168mm 1/32

印 张: 9.5

字 数: 240 千字

印 数: 1-500 册

版 次: 2009 年 4 月 第 1 版

印 次: 2009 年 4 月 第 1 次印刷

书 号: ISBN 978-7-203-06375-9

定 价: 20.00 元

如有印装质量问题请与本社联系调换

前 言

凡是学习了一种程序设计课程并能编写一些实用程序的人，也许都有这样一种体会，学会编程容易，但是想编出好程序难。一个高效率的程序不仅需要“编程小技巧”，更需要合理的数据组织和清晰高效的算法。算法设计与分析就是利用一些数学方法把数据结构中的操作程序化，使之解决实际问题，并能通过一些手段对算法的运行时间和存储空间作出分析与评价。因此，算法设计与分析是计算机科学与技术的一个核心问题。通过对计算机算法系统的学习与研究，使读者理解和掌握算法设计的主要方法，并利用这些方法去解决在计算机科学与技术中所遇到的各种问题，去设计计算机系统的各种软件中可能遇到的算法，并对所设计的算法作出科学的分析与评价。因此，算法设计与分析，不仅对计算机专业的技术人员，而且对使用计算机的其他专业技术人员，都是非常重要的。

本书以实例为线，循序渐进地说明算法分析的一般方法；然后以算法设计技术为纲，按照所叙述算法的设计思想、实现步骤、数据结构、算法的具体描述以及复杂性分析等几个方面，逐个介绍各种算法设计技术及其分析方法；最后又列举了常见的一些经典问题的解决方法。

全书分为三部分，第一部分为基础篇，旨在让读者明确什么是算法与算法分析，如何进行算法设计与分析，是算法设计与分析的入门内容。包括第1章至第3章。第1章介绍算法的基本概念及描述算法的方式方法。第2章介绍算法的评价，包括时间复杂性分析与空间复杂性分析。第3章介绍算法设计的基本工具及

优化方法。第二部分为核心篇，介绍在计算机科学与技术中常用的各类算法思想与策略，以经典问题为实例阐述各类算法的设计思想、设计步骤及算法复杂性分析。并对各类算法作出了比较与评估。包括第4章、第5章。第4章介绍了各类算法策略。具体有迭代算法，包括递推法和倒推法；蛮力算法，包括排序算法和枚举法及其他一些蛮力算法；分治算法，包括二分法及二分法的变异；贪心算法，包括哈夫曼编码、单源最短路径问题、最小生成树问题；动态规划策略，包括货郎担问题、资源分配问题、设备更新问题及最长公共子序列问题；回溯算法，包括n后问题和图着色问题；随机算法，包括Sherwood算法和Monte Carlo算法。第5章介绍图的搜索算法策略。具体包括图的深度优先遍历和广度优先遍历、网络流量的Ford_ Fulkerson方法和最大容量扩张方法及最短路径扩张方法、二分图的最大匹配问题等。第三部分为应用篇，列举了经典的算法设计与分析实例，旨在让读者从实例中发现算法设计的规律与技巧，加强实际应用能力。

本书所选内容适当，循序渐进，互相衔接，逐步展开。以实例说明算法设计与分析的重要性，并列举了经典例题，展现算法设计与分析的全过程，可以作为计算机专业技术人员及从事计算机工作的非专业人员及高等职业院校计算机专业学生的参考资料。

本书所有例子均采用C/C++描述语言作为表述手段，在保持C语言优点的同时，尽量使算法的描述简明、清晰。

最后，诚恳欢迎各位读者对本书的缺点、错误提出批评。

目 录

基 础 篇

第1章 算法的基本概念	3
1.1 算法的定义与特征	3
1.2 求解问题的一般过程	4
1.3 算法设计及其方法	6
1.3.1 算法设计应注意的问题	7
1.3.2 算法设计的基本方法	7
1.4 算法的描述方法	9
1.4.1 算法描述简介	9
1.4.2 算法描述约定	14
1.4.3 实例分析	17
第2章 算法分析基础	20
2.1 算法的分析体系	20
2.1.1 算法分析的评价	20
2.1.2 算法的时间复杂性分析	21
2.1.3 算法的空间复杂性分析	30
2.2 算法分析实例	31
2.3 NP 完全理论	39
第3章 算法基本工具和优化技巧	45
3.1 循环和递归	45
3.1.1 循环的设计	45
3.1.2 递归的设计	53

2 算法设计与分析

3.1.3 循环与递归的比较	56
3.2 算法与数据结构	64
3.2.1 常用的数据结构	64
3.2.2 利用数据结构构造趣味矩阵	76
3.3 算法优化基本技巧	80
3.3.1 算术运算的应用	80
3.3.2 标志位的应用	83
3.3.3 信息数字化	87
3.4 优化算法的实例模型	90
3.4.1 杨辉三角形的应用	92
3.4.2 最大公约数与最小公倍数的应用	94
3.4.3 猜数游戏	97
3.4.4 数字移位运用	100

核 心 篇

第4章 算法基本策略	105
4.1 迭代算法	105
4.1.1 递推法	105
4.1.2 倒推法	108
4.2 蛮力算法	113
4.2.1 排序算法	113
4.2.2 枚举法	121
4.3 分治算法	127
4.3.1 分治思想	127
4.3.2 二分法	129
4.3.3 二分法变异	136
4.4 贪心算法	143
4.4.1 贪心算法思想	143

4.4.2	哈夫曼编码	145
4.4.3	单源最短路径问题	148
4.4.4	最小生成树问题	149
4.5	动态规划	155
4.5.1	动态规划	155
4.5.2	货郎担问题	158
4.5.3	资源分配问题	162
4.5.4	设备更新问题	164
4.5.5	最长公共子序列问题	166
4.6	回溯算法	170
4.6.1	回溯算法思想	171
4.6.2	n 后问题	175
4.6.3	图的着色问题	176
4.7	随机算法	178
4.7.1	数值概率算法	180
4.7.2	舍伍德(Sherwood)算法	181
4.7.3	拉斯维加斯(Las Vegas)算法	184
4.7.4	蒙特卡罗(Monte carlo)算法	189
4.8	各种算法策略的比较	194
4.8.1	各种算法特点分析	194
4.8.2	算法侧重的问题类型	197
第5章	图的搜索算法	200
5.1	图的搜索概述	200
5.1.1	图及其术语	200
5.1.2	图的搜索及其术语	204
5.2	图的遍历	205
5.2.1	图的深度优先搜索遍历	206
5.2.2	图的广度优先搜索遍历	209

4 算法设计与分析

5.3 网络流量问题	213
5.3.1 网络流量引言	213
5.3.2 Ford_Fulkerson 方法和最大容量扩张	217
5.3.3 最短路径扩张问题	222
5.4 二分图的最大匹配问题	227
5.4.1 二分图引言	227
5.4.2 二分图最大匹配的匈牙利方法	230

应 用 篇

第6章 算法设计实践	241
6.1 递归与分治策略实例	241
6.2 贪心算法实例	251
6.3 动态规划算法实例	263
6.4 其他算法	274
6.4.1 货郎担问题的其他解法	274
6.4.2 背包问题的其他解法	281
参考文献	293



基
础
篇

第1章 算法的基本概念

计算机系统在任何软件，都是由大大小小的各种软件组成部分构成，各自按照特定的算法来实现，算法的好坏直接决定所实现软件性能的优劣。用什么方法来设计算法，所设计算法需要什么样的资源，需要多少运行时间、多少存储空间，如何判定一个算法的好坏，在实现一个软件时，都是必须予以解决的。计算机系统操作系统、语言编译系统、数据库管理系统以及各种各样的计算机应用系统中的软件，都必须用一个个具体的算法来实现。因此，算法设计与分析是计算机科学与技术的一个核心问题。

“算法”在中国最早出自《周髀算经》，而英文 algorithm 来自于9世纪波斯数学家比阿勒·霍瓦里松（阿尔·花拉子米）的名字 al-Khwarizmi，“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”。欧几里得曾在他的著作中描述过求两个数的最大公因子的过程。20世纪50年代，欧几里得所描述的这个过程，被称为欧几里得算法，欧几里得算法被人们认为是历史上第一个算法。从此，算法这个术语在学术上便具有了现在的含义。

1.1 算法的定义与特征

对于计算机科学来说，算法的概念至关重要。通俗地讲，算法就是对解决这个问题的方法和步骤的描述，是指令的有限序列。它有以下五个特性：

- (1) 输入：一个算法有零个或多个输入。
- (2) 输出：一个算法有一个或多个输出。
- (3) 有穷性：一个算法必须总是在执行有穷步之后结束，且每一步都在有穷时间内完成。
- (4) 确定性：算法中的每一条指令必须有确切的含义，对于相同的输入只能得到相同的输出。
- (5) 可行性：算法描述的操作可以通过已经实现的基本操作执行有限次来实现。

算法由操作、控制结构、数据结构三要素组成。操作包括算术运算：加、减、乘、除；关系运算：大于、小于、等于、不等于；逻辑运算：与、或；非数据传送：输入、输出、赋值。控制结构是指各操作之间的执行次序。比如：顺序结构、选择结构、循环结构。数据结构是指数据的存储方式及处理方式，它与算法设计是紧密相关的。

算法是计算机学科中最具有方法论性质的核心概念，也被誉为计算机学科的灵魂。

1.2 求解问题的一般过程

我们学习算法设计的重点就是把人类找到的求解问题的方法、步骤，以过程化、形式化、机械化的形式表示出来，以便让计算机执行。所以就把我们学习的目标定为“用计算机求解问题”。现实中，在解决一个问题时，根据不同的经验、不同的环境会采用不同的方法；用计算机解决现实中的问题，同样也有很多不同的方法，但解决问题的基本步骤是相同的。

下面给出用计算机求解问题的一般步骤：

1. 问题分析

准确、完整地理解和描述问题是解决问题的第一步。要做到

这一点，必须注意以下一些问题：在未经加工的原始表达中，所用的术语是否都明白其准确定义？题目提供了哪些信息？这些信息有什么用？题目要求得到什么结果？题目中做了哪些假定？是否有潜在的信息？判定求解结果所需要的中间结果有哪些？等等。针对每个具体的问题，必须认真审查问题描述，理解问题的真实要求。

2. 数学模型的建立

用计算机解决实际问题必须有合适的数学模型，因为在现实问题面前，计算机是无法对一个实际问题建立数学模型的。建立数学模型可以考虑这样两个基本问题：最适合于此问题的数学模型是什么？是否有已经解决了的类似问题可供借鉴？如果上述第二个问题的答复是肯定的，那么通过对类似问题的分析、比较和联想，可加速问题的解决。

3. 算法设计与选择

算法设计是指设计求解某一特定类型问题的一系列步骤，并且这些步骤是可以通过计算机的基本操作来实现的。算法设计要同时结合数据结构的设计，简单地说，数据结构的设计就是选取存储方式。算法的设计与模型的选择更是密切相关的，但同一模型仍然可以有不同的算法，而且它们的有效性可能有相当大的差距。

在这些步骤中，算法设计是解决问题的核心。

4. 算法分析

算法分析的目的，首先是为了对算法的某些特定输入估算该算法所需的内存空间和运行时间；其次是为了建立衡量算法优劣的标准，用以比较同一类问题的不同算法。通常将时间和空间的增长率作为衡量的标准。

5. 算法表示

对于复杂的问题，确定算法后可以通过图形准确表示算法。

算法的表示方式有很多,如算法流程图、盒图、PAD图和伪码(类似于程序设计语言)等。

6. 算法实现

根据选用的程序设计语言,解决下列问题:有哪些变量,它们是什么类型?需要多少数组,规模有多大?用什么结构来组织数据?需要哪些子算法?等等。

算法的实现方式,对运算速度和所需内存容量都有很大影响。

7. 程序调试

算法测试的实质是对算法应完成任务的实验证实,同时确定算法的使用范围。测试方法一般有两种:白盒测试和黑盒测试。白盒测试也称结构测试或逻辑驱动测试,它是对算法的各个分支进行测试;黑盒测试也称功能测试,它检验给定的输入是否有指定输出。

如何选择算法测试中的输入,还没有准确答案。通常采用的方法是,对输入数据做有代表性的采样,使之对被测试算法的各个语句、分支和路径尽可能都检查到。对输入集中的边界点也要进行测试。经测试验证正确的算法,在较大程度上可以相信它的正确性。

8. 结果整理文档编制

编制文档的目的是让人了解你编写的算法,增加程序的可读性。首先要把代码编写清楚。代码本身就是文档。同时还要采用注释的方式,说明代码中的数据结构与数据类型。另外还包括算法流程图、自顶向下各研制阶段的有关记录、算法正确性证明(或论述)、算法测试结果、对输入/输出的要求及格式的详细描述等。

1.3 算法设计及其方法

算法设计作为用计算机解决问题的一个步骤,其任务是对各

类具体问题设计良好的算法;作为一门课程,是研究设计算法的规律和方法。

1.3.1 算法设计应注意的问题

1. 正确性(Correctness)

一切合法的输入数据都能得出满足要求的结果;典型、苛刻的几组输入数据也能够得出满足要求的结果。

2. 可读性(Readability)

算法应该易于人的理解;晦涩难读的算法易于隐藏较多错误而难以调试。

3. 健壮性(Robustness)

当算法有异常情况时,处理出错的方法是返回一个表示错误或错误性质的值,以便在更高的抽象层次上进行处理。

4. 高效率与低存储量需求

效率指的是算法执行时间;存储量指的是算法执行过程中所需的最大存储空间。我们总希望代价最小,但价值最高。

1.3.2 算法设计的基本方法

算法设计的基本方法如下:

1. 结构化方法“自顶向下,逐步求精”

结构化方法总的指导思想是:自顶向下,逐步求精。它的基本原则是功能的分解与模块化。所谓“自顶向下”,是将现实世界的问题经抽象转化为逻辑空间或求解空间的问题。是将复杂且大的问题划分为较小问题,找出问题的关键和重点,然后抽象、概括地描述问题。所谓“逐步求精”,是将复杂问题经抽象化处理变为相对比较简单的问题。经若干步精化处理,最后细化到用“三种基本结构”及基本操作去描述算法。

结构化算法设计技术的优点: