



ACM-ICPC程序设计系列

算法设计与实现

● 主编 陈宇 吴昊



哈尔滨工业大学出版社
HARBIN INSTITUTE OF TECHNOLOGY PRESS



ACM-ICPC程序设计系列

算法设计与实现

- 主 编 陈 宇 吴 昊
- 副主编 张 健 朴秀峰
黄英来 郭继峰



哈爾濱工業大學出版社
HARBIN INSTITUTE OF TECHNOLOGY PRESS

内 容 简 介

本书系统地介绍了计算机算法设计的基本知识和常用方法,并根据具体的实例来编程实现,在注重算法基本知识的同时,突出了学习方法和实践技巧。全书共分 17 章,包括算法原理、数据结构基本知识、递归、高精度、贪心、动态规划、搜索、线段树、字符串、最小生成树、矩阵连乘、二分和枚举、母函数、树状数组、高斯消元、AC 自动机和哈希表,覆盖了计算机算法所需的知识点,并附有大量的应用实例。书中的代码规范、简洁、易懂,不仅能帮助读者理解算法原理,还能教会读者很多实用的编程技巧。

本书既可以作为大专院校信息与计算科学、计算机专业及信息安全专业的算法设计课教材,也可以作为大中学校 ACM 竞赛的培训教材,还可供计算机软硬件研发人员参考。

图书在版编目(CIP)数据

算法设计与实现/陈宇,吴昊,主编. —哈尔滨:
哈尔滨工业大学出版社,2014. 6
(ACM-ICPC 程序设计系列)
ISBN 978 - 7 - 5603 - 4724 - 0

I. ①算… II. ①陈… ②吴… III. ①电子计算机—
算法设计—高等学校—教材 IV. ①TP301.6

中国版本图书馆 CIP 数据核字(2014)第 093555 号

策划编辑 赵文斌 杜 燕
责任编辑 刘 瑶
出版发行 哈尔滨工业大学出版社
社 址 哈尔滨市南岗区复华四道街 10 号 邮编 150006
传 真 0451 - 86414749
网 址 <http://hitpress.hit.edu.cn>
印 刷 哈尔滨市工大节能印刷厂
开 本 787mm×960mm 1/16 印张 25 字数 527 千字
版 次 2014 年 6 月第 1 版 2014 年 6 月第 1 次印刷
书 号 ISBN 978 - 7 - 5603 - 4724 - 0
定 价 46.00 元

(如因印装质量问题影响阅读,我社负责调换)

ACM-ICPC 程序设计系列图书编委会

主 任 俞经善

副主任 金 博 陈 宇 孙大烈 孟繁军
殷明浩 朴秀峰

委 员 (按姓氏笔画排序)

丁 雨 马占飞 王 平 王 斌
王翠青 乔 付 邢海峰 刘丕娥
纪洪波 李 军 李 敏 杨明莉
迟呈英 周李涌 周治国 钟 辉
龚 丹 鲁静轩 滕国库

序 言

ACM-ICPC(ACM 国际大学生程序设计竞赛)被称为计算机领域的奥林匹克,是计算机领域高水平的智力角斗场。ACM-ICPC 于 1970 年起源于北美,由于涉及知识面广、注重实践,并有着公正的竞赛机制和颇有意思的比赛过程,因此在全球范围内迅速流行。其命题的考查范围除了程序设计、算法、数据结构、操作系统、计算机网络,编译原理等计算机专业相关学科知识外,离散数学、初等数论、组合数学、图论、计算几何等相关的数学知识也均包括在内,而且考查的知识范围还在持续扩大之中。

ACM-ICPC 程序设计系列图书是编者们在多年参与 ACM-ICPC 的过程中,对竞赛、训练的内容进行总结,从学科专题的角度集结而成。本套图书包含组合数学、初等数论、计算几何、图论和程序设计训练题解等内容,从竞赛的角度出发,介绍其中的基本概念和研究方法,主要以应用为目的。

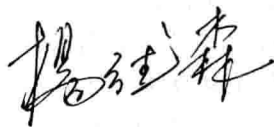
组合数学源远流长,古代的数学游戏和美学消遣,以无穷的魅力激发人们的聪明才智和数学兴趣。组合数学涉及内容广泛,知识点庞杂,与很多数学分支相交叉,主要研究将一些元素安排成种种集合的问题。组合数学被广泛应用在计算机科学、编码和密码学、物理、生物等学科以及交通管理、城市规划、企业管理等领域。初等数论是研究数的规律,特别是整数的性质的数学分支。初等数论就是用初等、朴素的方法去研究数论,是数论的一个最古老的分支。数论在计算机科学和应用数学的发展中得到了广泛的应用。比如,在计算方法、代数编码、组合论等方面都广泛使用了初等数论范围内的许多研究成果。计算几何于 20 世纪 70 年代末从算法设计与分析中分化而来,主要研究“几何图形信息(曲面和三维实体)的计算机表示、分析、修改和综合”。计算几何已经成长为一个被广泛认同的学科。在众多的应用领域(如计算机图形学、地理信息系统和机器人学等)都发挥着重要的作用。图论作为一个数学分支,有一套完整的体系和广泛的内容。它以图为研究对象,其应用范围非常广泛。不但应用于自然科学,且在社会科学中也有应用。当用图论来解决实际问题时,几乎都能引出复杂的图论模型,而这些模型一般情况下如果没有计算机的帮助很难分析出来,因此图论的快速发展和推广与计算机科学和信息科学的快速发展是分不开的。

本系列图书以介绍这些专题学科的基本概念和方法为基准,目的是使读者能在短时间内了解专题的主要内容。整套图书适用于参加初级、中级 ACM-ICPC 和信息学竞赛学生,对

计算机程序设计和算法感兴趣的读者同样有指导意义,同时对在该方向上有所研究的人士也有一定的参考意义,可作为竞赛代表队的培训教材,也可作为相关课程实践教学教材。

本系列图书是由 ACM-ICPC 中国·东北地区组织委员会组织策划,由哈尔滨工程大学、大连理工大学、东北大学、东北林业大学、东北师范大学等学校的老师(教练)群策群力共同完成。ACM-ICPC 中国·东北地区组织委员会一直关注着竞赛与实践教学的结合,使大学生们通过这个活动掌握更多的学科知识和提高分析问题、解决问题的能力。本系列图书的出版能够将一些有实际意义的专题学科知识以简洁而清晰的方式最快地介绍给大学生们,可以引起更多程序设计爱好者的兴趣,可以成为 ACM-ICPC 参赛队员攀登道路上的一块垫脚石。

ACM-ICPC 中国·东北地区组织委员会主席
第 34 届 ACM-ICPC 全球总决赛执行主席
哈尔滨工程大学副校长



2012 年 1 月

前 言

21 世纪是信息时代,作为信息科学和计算机科学的数学基础,算法设计受到越来越多的关注。纵观计算机学科数十年发展的历史,算法与计算复杂性理论一直是计算机科学研究的热点和活跃领域,也是获得图灵奖最多的研究领域之一。面对计算机应用领域的大量问题,最重要的是根据问题的性质选择正确的求解思路,即找到一个好的算法,特别在复杂的、海量信息的处理中,一个好的算法往往起着决定性的作用。

本书系统地介绍了算法设计的基本知识和算法设计常用方法,并根据具体的实例来编程实现,在注重常用算法基本知识的同时,突出了在线实现环节。全书共分 17 章,第 1 章介绍算法原理与 ACM 竞赛入门知识,包括在线评测系统的使用。第 2 章主要介绍了数据结构的基本知识,介绍线性表的顺序和链式存储结构,以及栈和队列在上述两种存储结构下的表示。第 3 章主要介绍递归的定义、特点和基本原理,在此基础上给出递归的一些性质,并对递归求解的问题进行了分类,最后通过实际的案例来讲解递归的算法实现。第 4 章介绍高精度计算的基本实现,并证明和阐述了 Stirling 公式的原理和应用。第 5 章介绍贪心算法的概念及其在实际中的应用。第 6 章主要介绍动态规划的基本概念和基本应用,并在此基础上阐述动态规划算法的基本特征和解题步骤。进一步对动态规划算法的基本问题进行了介绍,并重点讲解动态规划中的背包问题,详细介绍了其中的 0/1 背包问题。第 7 章主要介绍搜索的核心思想、搜索通常的实现方式、搜索的优缺点以及搜索解题的一般思路。第 8 章主要介绍线段树的基本结构和特点,及其具体实现。并针对线段树的基本结构和操作展开了细致的讨论和例题解析。第 9 章主要介绍 KMP 算法的原理和 KMP 算法的应用,进一步阐述了 Trie 树的基本原理、性质和相关操作,并对 Trie 树的复杂度进行了分析。第 10 章主要介绍图论中的最小生成树的基本概念和基本应用,以及最短路中的 Dijkstra 算法、Bellman-Ford 算法和 Floyd 算法的基本思路 and 具体编程实现。第 11 章主要介绍 Fibonacci 数列和矩阵乘法的基本原理及其典型例题,同时介绍将 Fibonacci 数列和矩阵乘法相结合的题型的解题思路。第 12 章在介绍二分和枚举的基本定义和原理的基础上,阐述二分的基本特征,并说明二分算法的实现步骤。第 13 章介绍普通母函数的基本概念及应用,同时介绍指数型母函数的概念及应用。第 14 章主要介绍树状数组的建立以及在此基础上的实现,并针对其中的重点部分进行实际例题的分析和解题过程的讲解。第 15 章介绍高斯消元的基本原理以及针对其在 ACM 竞赛中的典型应用进行分析。第 16 章主要介绍 AC 自动机的概

述,并重点介绍 AC 自动机的构造过程,同时针对其构造过程进行典型例题的分析和讲述。第 17 章介绍哈希表的由来以及哈希函数的主要构造和选取方法,并重点介绍在哈希表中处理冲突的方法和具体实现过程。

书中的代码规范、简洁、易懂,不仅能帮助读者理解算法原理,还能教会读者很多实用的编程技巧。本书既可以作为大专院校信息与计算科学、计算机专业及信息安全专业的算法设计课教材,也可以作为大中学校 ACM 竞赛的培训教材,还可供计算机软件研发人员参考。

本书由陈宇统稿。具体分工如下:第 1、2 章由吴昊编写,第 3~5 章由朴秀峰编写,第 6~8 章由张健编写,第 9~11 章由黄英来编写,第 12~15 章由陈宇编写,第 16、17 章由郭继峰编写。

本教材在编写中参考了很多国内相关文献资料,同时得到了东北林业大学信息学院领导、东北地区 ACM 竞赛指导委员会和哈尔滨工业大学出版社的大力支持。同时感谢东北林业大学 ACM 集训队的队员们对本书的算法进行的测试,他们为本书的出版付出了辛勤的劳动,做出了很大的贡献。

由于时间和水平所限,书中难免存在不妥和错漏之处,欢迎同仁或读者指正。如果在阅读中发现问题,请通过书信或电子邮件告诉我们,我们期望读者对本书提出建设性意见,以便修订再版时改进。读者可以通过以下方式与我联系:

电子邮箱:nefu_chenyu@163.com

编 者

2014 年 3 月于哈尔滨

目 录

第 1 章 算法原理与 ACM 入门	(1)
1.1 算法概述	(1)
1.2 在线测评系统的使用	(3)
1.3 编程基本知识	(18)
第 2 章 数据结构的基本知识	(22)
2.1 线性表	(22)
2.2 栈	(27)
2.3 队列	(33)
2.4 串	(37)
2.5 抽象数据类型串的实现	(39)
第 3 章 递归	(53)
3.1 递归的定义	(53)
3.2 递归的实际应用	(57)
3.3 习题	(66)
第 4 章 高精度计算	(67)
4.1 Stirling 公式	(67)
4.2 高精度运算	(70)
4.3 习题	(91)
第 5 章 贪心	(92)
5.1 贪心的原理	(92)
5.2 贪心算法及其应用	(94)
5.3 习题	(115)

第 6 章 动态规划	(116)
6.1 动态规划的原理	(116)
6.2 动态规划的应用	(117)
6.3 背包问题	(140)
6.4 习题	(145)
第 7 章 搜索	(146)
7.1 枚举	(146)
7.2 深度优先搜索	(150)
7.3 广度优先搜索	(160)
7.4 习题	(171)
第 8 章 线段树	(172)
8.1 引言	(172)
8.2 线段树的引用	(172)
8.3 线段树的基本结构及特点	(172)
8.4 线段树的具体实现	(173)
8.5 例题解析	(176)
8.6 习题	(186)
第 9 章 字符串	(188)
9.1 KMP 算法	(188)
9.2 Tire 树	(201)
9.3 习题	(212)
第 10 章 最小生成树和最短路	(213)
10.1 最小生成树	(213)
10.2 最短路	(223)
10.3 习题	(237)
第 11 章 矩阵连乘	(238)
11.1 初识 Fibonacci 数列	(238)

11.2	Fibonacci 数列的应用	(239)
11.3	矩阵乘法的基本原理	(243)
11.4	Fibonacci 数列与矩阵连乘	(246)
11.5	矩阵乘法的拓展	(255)
11.6	习题	(263)
第 12 章	二分和枚举	(264)
12.1	枚举概述	(264)
12.2	二分概述	(269)
12.3	二分的实际应用	(273)
12.4	习题	(287)
第 13 章	母函数	(288)
13.1	母函数概述	(288)
13.2	指数型母函数	(301)
13.3	习题	(306)
第 14 章	树状数组	(307)
14.1	树状数组的基本原理	(307)
14.2	树状数组的应用	(310)
14.3	多维树状数组	(321)
14.4	习题	(329)
第 15 章	高斯消元	(330)
15.1	高斯消元的原理	(330)
15.2	高斯消元的应用	(336)
15.3	习题	(347)
第 16 章	AC 自动机	(348)
16.1	AC 自动机概述	(348)
16.2	AC 自动机的构造	(348)
16.3	AC 自动机的原理	(348)
16.4	AC 自动机的应用	(351)

16.5 习题.....	(367)
第 17 章 哈希表	(368)
17.1 哈希表概述.....	(368)
17.2 哈希函数的构造方法.....	(370)
17.3 解决哈希冲突的方法.....	(373)
17.4 习题.....	(386)
参考文献	(387)

第 1 章 算法原理和 ACM 入门

本章要点

本章主要介绍关于算法的相关概念及算法时间复杂度的计算;同时介绍在线测评系统的五种输入形式和三种输出形式,以及在线测评系统测评的结果表达;还介绍一些简单的编程基本知识及基本的数学问题.

1.1 算法概述

1.1.1 基本定义

定义 1.1 算法分析(Algorithm Analysis)是对算法所需要的时间和空间两种计算机资源进行估算.算法分析即是分析程序执行的时间复杂性(Time Complexity)和空间复杂性(Space Complexity).

定理 1.1 算法分析的目的:

①设计算法,设计出复杂性尽可能低的算法;②选择算法,在多种算法中选择其中复杂性最低者.

定理 1.2 评价算法的 3 条主要标准:

①算法实现所耗费的时间;②算法实现所耗费的存储空间,其中主要考虑辅助存储空间;③算法应易于理解,易于编码,易于调试等.

定理 1.3 算法效率的衡量方法:

①事后统计法;②事前分析估算法.

定理 1.4 和算法执行时间相关的因素:

①问题中数据存储的数据结构;②算法采用的数学模型;③算法设计的策略;④问题的规模;⑤实现算法的程序设计语言;⑥编译算法产生的机器代码的质量;⑦计算机执行指令的速度.

定义 1.2 伪代码(Pseudocode)是介于自然语言和程序设计语言之间的方法,它采用某一程序设计语言的基本语法,操作指令可以结合自然语言来设计.

算法的描述概况见表 1.1.

表 1.1 算法的描述概况

算法的描述语言	优点	缺点	使用方法	注意事项
自然语言	容易理解	冗长、二义性	粗线条描述 算法思想	避免写成自然段
流程图	流程直观	缺少严密性、 灵活性	描述简单算法	注意抽象层次
程序设计语言	能由计算机执行	抽象性差, 对语言要求高	算法需要验证	将算法写成子函数
伪代码	表达能力强,抽象 性强,容易理解			

1.1.2 算法时间复杂度的计算

定义 1.3 算法的空间复杂度是指执行这个算法所需要的内存空间.

定义 1.4 算法的时间效率是问题规模的函数. 设随着问题规模 n 的增长, 算法执行时间的增长率和 $f(n)$ 的增长率相同, 则可记作: $T(n) = O(f(n))$, 称 $T(n)$ 为算法的渐近时间复杂度 (Asymptotic Time Complexity), 简称算法的时间复杂度. 其中 O 是数量级的符号.

定义 1.5 算法的运行时间是指一个算法中所有语句的频度之和.

例 1.1 计算此算法的运行时间.

```
for(j=1;j<=n;++j)
    for(k=1;k<=n;++k)
        ++x;
```

解 语句“ $++x$ ”“ $k<=n; ++k$ ”的频度是 n^2 , 语句“ $j=1$ ”“ $k=1$ ”的频度是 1, 语句“ $j<=n; ++j$ ”的频度是 n .

算法运行时间为 $3n^2 + 2n + 2$.

定理 1.5 当一个算法的算法运行时间为 $n^2 + n + 1$, 由于 $n^2 + n + 1$ 与 n^2 的数量级相等 (该表达式当 n 足够大时约等于 n^2), 就说这个算法的渐进时间复杂度为 $T(n) = O(n^2)$.

定理 1.6 算法时间复杂度一般均表示为以下几种数量级的形式 (n 为问题的规模, c 为一常量): $O(1)$ 称为常数级; $O(\log n)$ 称为对数级; $O(n)$ 称为线性级; $O(nc)$ 称为多项式级; $O(c^n)$ 称为指数级; $O(n!)$ 称为阶乘级.

例 1.2 分析算法的时间复杂度.

```
Temp=i;i=j;j=temp;
```

以上 3 条单个语句的频度均为 1, 该算法段的执行时间是一个与问题规模 n 无关的常数. 算法的时间复杂度为常数级, 记作 $T(n) = O(1)$.

说明 如果算法的执行时间不随着问题规模 n 的增加而增长,即使算法中有上千条语句,其执行时间也不过是一个较大的常数.此类算法的时间复杂度是 $O(1)$.

例 1.3 计算算法的时间复杂度.

```
x=0;y=0;
for(k=1;k<=n;k++)
    x++;
for(i=1;i<=n;i++)
    for(j=1;j<=n;j++)
        y++;
```

该算法段的时间复杂度为 $T(n) = O(n^2)$.

说明 当有若干个循环语句时,算法的时间复杂度是由嵌套层数最多的循环语句中最内层语句的频度 $f(n)$ 决定的.

例 1.4 计算算法的时间复杂度.

```
x=1; .....1
for(i=1;i<=n;i++) .....2
    for(j=1;j<=i;j++) .....3
        for(k=1;k<=j;k++) .....4
            x++; .....5
```

该算法段中频度最大的语句是 5,从内层循环向外层分析语句 5 的执行次数,则该算法段的时间复杂度为 $T(n) = O(n^3)$.

1.2 在线测评系统的使用

由于 ACM 竞赛题目的输入数据和输出数据一般有多组,并且格式多种多样,所以,如何处理题目的输入输出是使用在线测评系统的一项最基本的要求.

1.2.1 输入的 5 种形式

例 1.5 对于 $a+b$ 的输入和输出练习(1). (Hdu1089)

你的任务是计算 $a+b$.

☐输入

输入由一系列的一组整数 a 和 b 组成, a, b 中间用一个空格分隔开,每组整数组占一行.

☞ 输出

对于每组输入整数 a 和 b , 在一行输出 a 和 b 的和, 每行输入下一行紧跟着下一行输出.

☞ 输入样例

1 5

10 20

☞ 输出样例

6

30

☞ 分析

输入不说明有多少组输入, 以 EOF 为结束标志.

☞ 代码

```
#include<iostream>
using namespace std;
int main(int argc, char * argv[ ])
{
    int a, b;
    while( cin>>a>>b)
        cout<<a+b<<endl;
    return 0;
}
```

说明

(1) 此类输入解决方案.

C 语法:

```
while( scanf("%d %d", &a, &b) != EOF)
```

```
{
```

```
...
```

```
}
```

C++语法:

```
while( cin>>a>>b)
```

```
{
```

```
...
```

```
}
```

(2) Scanf 函数返回值就是读出的变量个数, 如: `scanf("%d %d", &a, &b);`.

如果只有一个整数输入, 返回值是 1; 如果有两个整数输入, 返回值是 2; 如果一个都没

有,则返回值是-1.

(3) EOF 是一个预定义的常量,等于-1.

例 1.6 对于 $a+b$ 的输入和输出练习(2). (Hdu1090)

你的任务是计算 $a+b$.

☞输入

输入包含第一行一个整数 N ,接下来有 N 行,每行包含一组整数 a 和 b , a,b 中间用一个空格分隔开,每组整数组占一行.

☞输出

对于每组输入整数 a 和 b ,在一行输出 a 和 b 的和,每行输入下一行紧跟着—行输出.

☞输入样例

2

1 5

10 20

☞输出样例

6

30

☞分析

输入一开始就会说输入组个数 N ,下面接着是 N 组输入中每组要输入的整数 a 和 b .

☞代码

```
#include<iostream>
using namespace std;
int main(int argc,char * argv[ ])
{
    int N,i,a,b;
    cin>>N;
    for(i=0;i<N;i++)
    {
        cin>>a>>b;
        cout<<a+b<<endl;
    }
}
```

说明

此类输入解决方案如下.