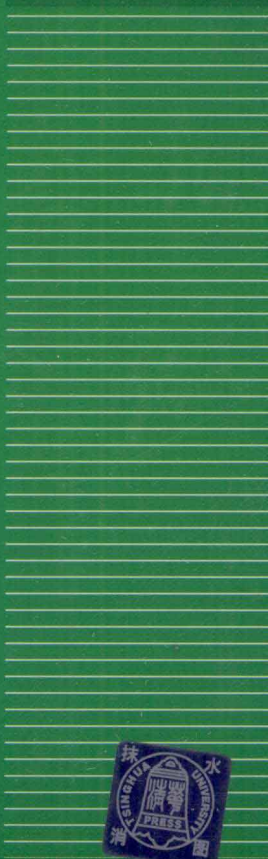
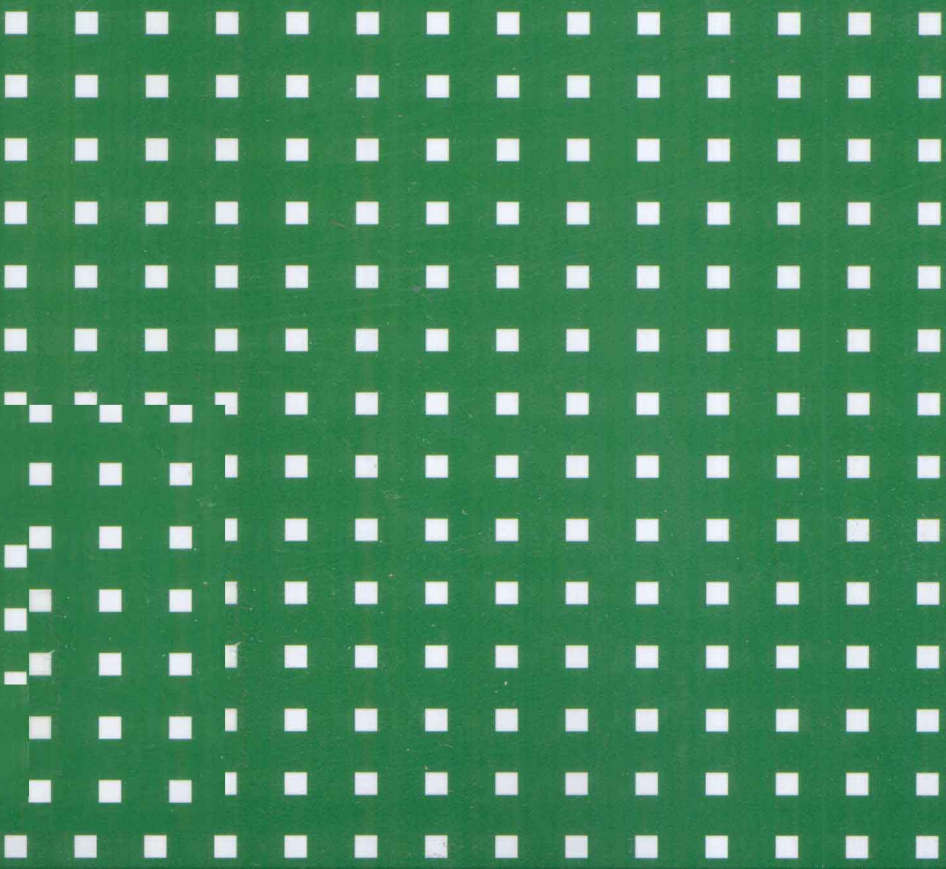


高等学校计算机专业教材精选·算法与程序设计

计算机常用算法与 程序设计案例教程

杨克昌 编著



清华大学出版社

高等学校计算机专业教材精选·算法与程序设计

计算机常用算法与 程序设计案例教程

杨克昌 编著

清华大学出版社
北京

内 容 简 介

本书遵循“精选案例,面向设计,深入浅出,注重能力培养”的要求,以“案例”形式实现“算法与程序设计”教学。本书精选枚举、递推、递归、回溯、动态规划、贪心算法与模拟等常用算法,精选各算法求解的典型案例,注重算法设计与程序实现,注重算法改进与程序优化,力求理论与实际相结合,算法与程序相统一。每一个案例求解,从案例提出、算法设计与描述,到程序实现、运行结果与讨论,环环相扣,融为一体。

书中所有案例求解给出详细的算法描述与完整的 C 程序。每章最后附有习题,在附录中给出习题求解提示,所有源程序可从清华大学出版社网站下载。

本书可作为高等院校计算机及相关专业“算法设计与分析”、“计算机程序设计”课程教材,也可供软件设计人员与计算机爱好者学习参考。

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

计算机常用算法与程序设计案例教程/杨克昌编著. —北京:清华大学出版社,2011.7
(高等学校计算机专业教材精选·算法与程序设计)

ISBN 978-7-302-25308-2

I. ①计… II. ①杨… III. ①电子计算机—算法理论—高等学校—教材 ②程序设计—高等学校—教材 IV. ①TP301.6 ②TP311.1

中国版本图书馆 CIP 数据核字(2011)第 068019 号

责任编辑:白立军 顾 冰

责任校对:焦丽丽

责任印制:李红英

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:010-62795954, jsjic@tup.tsinghua.edu.cn

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

印 装 者:北京鑫海金澳胶印有限公司

经 销:全国新华书店

开 本:185×260

印 张:19.75

字 数:473 千字

版 次:2011 年 7 月第 1 版

印 次:2011 年 7 月第 1 次印刷

印 数:1~3000

定 价:29.80 元

产品编号:039175-01

出版说明

我国高等学校计算机教育近年来迅猛发展,应用所学计算机知识解决实际问题,已经成为当代大学生的必备能力。

时代的进步与社会的发展对高等学校计算机教育的质量提出了更高、更新的要求。现在,很多高等学校都在积极探索符合自身特点的教学模式,涌现出一大批非常优秀的精品课程。

为了适应社会的需求,满足计算机教育的发展需要,清华大学出版社在进行大量调查研究的基础上,组织编写了《高等学校计算机专业教材精选》。本套教材从全国各高校的优秀计算机教材中精挑细选了一批很有代表性且特色鲜明的计算机精品教材,把作者们对各自所授计算机课程的独特理解和先进经验推荐给全国师生。

本系列教材特点如下。

(1) 编写目的明确。本套教材主要面向广大高校的计算机专业学生,使学生通过本套教材,学习计算机科学与技术方面的基本理论和基本知识,接受应用计算机解决实际问题的基本训练。

(2) 注重编写理念。本套教材作者群为各高校相应课程的主讲,有一定经验积累,且编写思路清晰,有独特的教学思路和指导思想,其教学经验具有推广价值。本套教材中不乏各类精品课配套教材,并努力把不同学校的教学特点反映到每本教材中。

(3) 理论知识与实践相结合。本套教材贯彻从实践中来到实践中去的原则,书中的许多必须掌握的理论都将结合实例来讲,同时注重培养学生分析问题、解决问题的能力,满足社会用人要求。

(4) 易教易用,合理适当。本套教材编写时注意结合教学实际的课时数,把握教材的篇幅。同时,对一些知识点按教育部教学指导委员会的最新精神进行合理取舍与难易控制。

(5) 注重教材的立体化配套。大多数教材都将配套教师用课件、习题及其解答,学生上机实验指导、教学网站等辅助教学资源,方便教学。

随着本套教材陆续出版,我们相信它能够得到广大读者的认可和支持,为我国计算机教材建设及计算机教学水平的提高,为计算机教育事业的发展做出应有的贡献。

清华大学出版社

前言

计算机算法与程序设计是计算机科学与技术的核心,是大学计算机相关专业的一门重要的专业基础课。通过对现有计算机本科专业“算法设计与分析”课程效果的调研分析,很多学生对学过的算法思路不甚了解,描述不清楚,设计不到位,无法应用算法设计程序解决一些常见的实际应用问题。缺少适合计算机本科层次的“算法与程序设计”教材是造成这一局面的重要原因之一。

一般现有“算法设计与分析”教材在算法选取上贪多求全、贪广求深,混杂一些难度大、理论深、应用少的带学术研究性质、适宜研究生阶段的算法内容。同时普遍存在算法与数据结构结合多,算法与程序设计结合少;罗列算法多、应用算法设计实际问题少。算法与程序设计脱节,算法理论与实际应用脱节,不利于算法设计的应用推广,也难以培养和提高学生应用算法与程序设计解决实际问题的能力。

为此,我们在编写《计算机程序设计经典题解》(清华大学出版社,2007.12)与《计算机常用算法与程序设计教程》(“十一五”国家级规划教材,人民邮电出版社,2008.11)的基础上进行优化整合,推出适合本科“算法与程序设计”教学实际的“案例”教程。

本教程遵循“精选案例,面向设计,深入浅出,注重能力培养”的宗旨编写,在常用算法应用案例的选取与深度的把握上,在算法理论与案例求解的结合上进行精心设计,力图适合高校计算机本科教学目标与知识结构的要求,体现以下五个特色。

(1) 首创“案例”形式实现算法与程序设计教学。

学习算法与程序设计是为了培养提高学生应用算法与程序设计解决实际问题的能力,算法与程序设计课程教学无疑是最适宜以“案例”形式来实现的。通过典型案例来引导算法设计的逐步深入,来展开程序设计的求解实施,实现以典型案例支撑算法,以算法设计指导案例求解的良性循环。

采用“案例”形式实现算法与程序设计教学在全国是首创,从案例的提出、算法设计与描述、程序设计与实现,到案例结果的讨论与变通,环环相扣,融为一体,学生看得见,摸得着,学得会,用得上,容易收到立杆见影、举一反三的效果。

(2) 注重常用算法的选取与组织。

在常用算法的选取上克服贪多求全、贪广求深,去除一些难度大、理论深、应用少的带学术研究性质的算法内容,结合本科教学实际与应用需求,选取枚举、递推、递归、回溯、动态规划、贪心算法与模拟等常用算法。其中模拟算法中的“竖式乘除模拟”是我们总结推广用于数论高精计算的创新成果。

对精选的各种常用算法,在介绍算法的基本理论与设计思路后,从实际案例的解决入手,讲述算法中要求本科学生掌握的基本思路、设计与实施步骤,避免出现本科阶段与研究生阶段的教学内容混杂不分,避免出现蜻蜓点水、面面俱到、空洞而不着边际的局面。

(3) 注重典型案例的精选与提炼。

针对选取的每一种常用算法,精选典型的实际案例,包括典型的数值求解,常见的数据

处理,有趣的智力测试,巧妙的模拟探索,既有引导入门的基础案例,也有难度较大的综合案例,既有创新趣题,也有经典名题,难度适宜,深入浅出。

培养学生的学习兴趣,激发学生的学习热情,不是一两句空洞说教所能奏效的,必须通过一些有趣的实际案例来引导。本教程针对所精选的常用算法,设计了初等难度基础型、中等难度应用型、较高难度综合型3种梯度的案例。这些案例的精选与提炼,有利于提高学生学习算法与程序设计的兴趣,有利于学生在计算机实际案例求解上开阔视野,使之在算法思路的开拓与设计技巧的应用上有一个深层次的锻炼与提高。其中难度较大的综合案例可作为课后的专题研究与课程设计选用。

(4) 注重算法设计与程序实现的紧密结合。

算法与程序实际上是一个统一体,不应该也不可能将它们对立与分割。本教程在材料的组织上克服了罗列算法多、应用算法设计实际问题少,算法与程序设计脱节、算法理论与实际应用脱节的问题。在讲述每一种常用算法的基本思路与设计步骤的基础上,落实到每一个案例求解,从案例的提出到算法设计与描述、从程序实现到案例结果的讨论与分析,环环相扣,融为一体,力求理论与实际相结合、算法与程序相统一,突出算法在解决实际案例中的核心地位与引导作用,切实提高对所学算法的理解和掌握。

教程采用功能丰富、应用面广、高校学生使用率最高的C语言描述算法、编写程序。为使读者使用方便,所有程序均在VC++ 6.0编译通过。

(5) 注重算法改进与程序优化。

教程对一些典型案例应用多种不同的算法设计,编写不同表现形式与不同设计风格的程序,体现了算法与程序设计的灵活性和多样性。

算法与程序设计都不是一成不变的,可以实施多层次多方位的变通,变通出成果,变通长能力。算法改进与程序优化的过程,既是提高案例求解效率的过程,也是算法设计能力培养与提高的过程,更是优化意识与创新能力增强的过程。

为方便读者程序设计练习与查阅,附录中提供各章习题求解提要,简介在VC++ 6.0环境下运行C程序的方法,并列出C常用函数。同时,教程中的所有源程序与各章习题的完整求解代码均可在指定网站下载。

本教程适合各高校计算机相关本科专业“算法与程序设计”课程教学。可供各级程序设计选拔赛与国际大学生程序设计竞赛(ACM)复习参考,也可供IOI、NOI及各省程序设计竞赛培训选用。

在书稿的编著过程中,湖南理工学院王岳斌教授、周持中教授、吴小明副教授、严权锋副教授与谭用秋博士等给予多方面的支持与协助,刘志辉硕士阅读了全部书稿并运行了所有程序,笔者在此一并深表感谢。

尽管每一案例求解都经反复核实检查,每一求解程序都经多次运行调试,因涉及内容较广,难免存在各种差错,恳请读者指正。

杨克昌

2011.1 于岳阳南湖

目 录

第 1 章 算法与程序设计概述	1
1.1 算法及其描述	1
1.1.1 算法定义	1
1.1.2 算法描述	3
1.2 算法的复杂性分析	7
1.2.1 时间复杂度	7
1.2.2 空间复杂度	12
1.3 算法与程序设计	13
1.3.1 算法与程序	13
1.3.2 结构化程序设计	18
习题 1	20
第 2 章 枚举	22
2.1 枚举概述	22
2.2 统计与求和	23
2.2.1 指定特殊整数	23
2.2.2 最简真分数	25
2.3 解方程	26
2.3.1 解佩尔方程	26
2.3.2 解超越方程	28
2.4 解不等式	30
2.4.1 分数不等式	30
2.4.2 代数和不等式	31
2.5 求最值	34
2.5.1 基于素数的代数和	34
2.5.2 整数的因数比	35
2.6 数组与数列	36
2.6.1 双和数组	36
2.6.2 基于 $2x+3y$ 的递推数列	38
2.7 数式探求	39
2.7.1 逆序乘积式	39
2.7.2 完美综合式	40
2.8 趣味数阵	43
2.8.1 素数幻方	43

2.8.2 和积三角形	46
2.9 枚举应用小结	48
习题 2	53
第 3 章 递推	55
3.1 递推概述	55
3.1.1 递推算法	55
3.1.2 递推实施步骤与描述	56
3.2 递推数列	58
3.2.1 摆动数列	58
3.2.2 分数数列	59
3.3 幂序列	60
3.3.1 双幂序列	60
3.3.2 幂积序列	62
3.4 数阵探索	67
3.4.1 杨辉三角	67
3.4.2 折叠方阵	69
3.5 整数划分问题	71
3.5.1 整数划分递推设计	71
3.5.2 整数划分递推优化	73
3.6 水手分椰子问题	74
3.6.1 水手分椰子	74
3.6.2 n 个水手分椰子	77
3.7 猴子爬山	79
3.7.1 简单案例的具体递推	79
3.7.2 一般情形的分级递推	80
3.8 递推应用小结	82
习题 3	83
第 4 章 递归	85
4.1 递归概述	85
4.2 排队购票	87
4.3 汉诺塔问题	89
4.3.1 求移动次数	90
4.3.2 展示移动过程	91
4.4 旋转数阵	92
4.4.1 双转向旋转方阵	92
4.4.2 m 行 n 列顺转矩阵	95
4.5 快速排序与选择	97

4.5.1	快速排序	97
4.5.2	分区交换选择	100
4.6	排列组合的实现	102
4.6.1	实现排列 $A(n, m)$	102
4.6.2	实现组合 $C(n, m)$	105
4.6.3	实现复杂排列	107
4.7	整数的拆分	109
4.7.1	拆分零数取自连续区间	109
4.7.2	拆分零数取自指定整数	111
4.8	递归应用小结	112
习题 4		115
第 5 章	回溯法	117
5.1	回溯法概述	117
5.1.1	回溯的概念	117
5.1.2	回溯描述	117
5.2	桥本分数式	122
5.2.1	桥本分数式概述	122
5.2.2	10 数字分数式	124
5.3	直尺与串珠	125
5.3.1	古尺神奇	126
5.3.2	数码串珠	128
5.4	逐位整除数探索	130
5.4.1	高逐位整除数	130
5.4.2	低逐位整除数	132
5.5	环序列	134
5.5.1	素数和环	134
5.5.2	德布鲁金环	136
5.6	装错信封问题	138
5.6.1	伯努利装错信封问题	138
5.6.2	特殊错位探索	141
5.7	别出心裁的情侣拍照	144
5.7.1	逐位安排与回溯	144
5.7.2	成对安排与回溯	146
5.8	回溯应用小结	148
习题 5		150
第 6 章	动态规划	152
6.1	动态规划概述	152

6.1.1	动态规划的概念	152
6.1.2	动态规划实施步骤	153
6.2	最长子序列探索	154
6.2.1	最长非降子序列	154
6.2.2	最长公共子序列	157
6.3	最优路径搜索	160
6.3.1	点数值三角形的最优路径	160
6.3.2	边数值矩形的最优路径	162
6.4	装载问题	165
6.5	0-1 背包问题	169
6.5.1	一般 0-1 背包问题	169
6.5.2	二维约束 0-1 背包问题	173
6.6	插入乘号问题	178
6.6.1	动态规划求解	178
6.6.2	基于组合枚举求解	181
6.7	动态规划应用小结	182
	习题 6	185
第 7 章	贪心算法	186
7.1	贪心算法概述	186
7.2	删数字问题	187
7.3	埃及分数式	190
7.3.1	选择最小分母构建	190
7.3.2	贪心选择范围的扩展	192
7.4	可拆背包问题	193
7.5	数列操作与极差	195
7.5.1	数列操作	195
7.5.2	数列操作优化	196
7.5.3	数列极差	198
7.6	哈夫曼树及其应用	201
7.6.1	哈夫曼树	201
7.6.2	哈夫曼编码	204
7.7	贪心算法应用小结	207
	习题 7	208
第 8 章	模拟	210
8.1	竖式乘除模拟	210
8.1.1	竖式除模拟	210
8.1.2	竖式乘模拟	211

8.2 乘数探求	211
8.2.1 积为若干个 1 构成	212
8.2.2 积为若干个 2011 构成	213
8.2.3 积的任意指定构成	214
8.3 尾数前移问题	215
8.3.1 限 1 位尾数前移	215
8.3.2 多位尾数前移	217
8.4 阶乘幂与排列组合数的计算	218
8.5 圆周率 π 的高精度计算	220
8.6 蒙特卡罗模拟计算	222
8.7 模拟发桥牌	224
8.8 泊松分酒	227
8.9 模拟应用小结	230
习题 8	231
第 9 章 算法的综合应用	232
9.1 最大子段和问题	232
9.1.1 序列的最大子段和	232
9.1.2 环序列的最大子段和	235
9.2 高斯皇后问题	238
9.2.1 高斯八皇后问题	238
9.2.2 n 皇后问题	240
9.2.3 皇后全控棋盘问题	243
9.3 马步遍历与哈密顿圈	248
9.3.1 马步遍历	249
9.3.2 马步型哈密顿圈	256
9.3.3 组合型哈密顿圈	260
9.4 算法的综合应用小结	266
习题 9	267
附录	268
附录 A 部分习题求解提要	268
附录 B 在 VC++ 6.0 环境下运行 C 程序方法简介	290
附录 C C 常用库函数	295
参考文献	299

第1章 算法与程序设计概述

1.1 算法及其描述

在计算机科学与技术中,算法(algorithm)是一个常用的基本概念。本节论述算法的定义与算法的描述。

1.1.1 算法定义

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方法。算法是程序设计的基础,是计算机科学的核心。计算机科学家哈雷尔在《算法学——计算的灵魂》一书中指出:“算法不仅是计算机学科的一个分支,它更是计算机科学的核心,而且可以毫不夸张地说,它和绝大多数科学、商业和技术都是相关的。”

在计算机应用的各个领域,技术人员都在使用计算机求解他们各自专业领域的课题,他们需要设计算法,编写程序,开发应用软件,所以学习算法对于越来越多的人来说变得十分必要。我们学习算法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,从而解决更多的实际问题。

下面首先给出算法的定义。

算法是计算机解决问题的过程,是解决某一问题的运算序列,或者说算法是问题求解过程的运算描述。当面临某一问题时,需要找到用计算机解决这个问题方法与步骤,算法就是解决这个问题的方法与步骤的描述。

1. 算法的三要素

算法由操作、控制结构与数据结构三要素组成。

(1) 操作

操作包括以下内容。

- 加、减、乘、除等算术运算;
- 大于、小于、等于、大于等于、小于等于、不等于等关系运算;
- 与、或、非等逻辑运算;
- 输入、输出、赋值等操作。

(2) 控制结构

控制结构包括以下内容。

- 顺序结构:各操作依次执行;
- 选择结构:由条件是否成立来选择操作执行;
- 循环结构:重复执行某些操作,直到满足某一条件才结束;
- 模块调用:一个模块调用另一个模块(包括自身直接或间接调用的递归结构)。

(3) 数据结构

算法的处理对象是数据,数据之间的逻辑关系、数据的存储方式与处理方式就是数据的

数据结构。

2. 算法的基本特征

一个算法由有限条可完全机械地执行的、有确定结果的指令组成。指令正确地描述了要完成的任务和它们被执行的顺序。计算机按算法所描述的顺序执行算法的指令能在有限的步骤内终止,或终止于给出问题的解,或终止于指出问题对此输入数据无解。

算法是满足下列特性的指令序列。

(1) 确定性

组成算法的每条指令是清晰的,无歧义的。

在算法中不允许有诸如“ $x/0$ ”之类的运算,因为其结果不能确定;也不允许有“ x 与 1 或 2 相加”之类的运算,因为这两种可能的运算应执行哪一个,并不确定。

(2) 可行性

算法中的运算是能够实现的基本运算,每一种运算可在有限的时间内完成。

在算法中两个实数相加是可行的;两个实数相除,例如求 $2/3$ 的值,在没有指明位数时需由无穷个十进制位表示,并不可行。

(3) 有穷性

算法中每一条指令的执行次数有限,执行每条指令的时间有限。

如果算法中的循环步长为零,运算进入无限循环,这是不允许的。

(4) 算法有零个或多个输入

算法所能接受的数据输入。有些输入数据需要在算法执行过程中输入,有些算法看起来没有输入,实际上输入已被嵌入算法之中。

(5) 算法有一个或多个输出

输出一个或多个与输入数据有确定关系的量,是算法对数据进行运算处理的结果。

通常求解一个问题可能会有多种算法可供选择,选择的主要标准是算法的正确性和可靠性,其次是算法所需要的存储空间少和执行时间短等。

3. 算法的重要意义

有人也许会认为:今天计算机运算速度这么快,算法还重要吗?

诚然,计算机的计算能力每年都在飞速增长,价格在不断下降。可不要忘记,日益发展的存储手段使我们需处理的信息量也在快速增长,Internet 的信息流量更在爆炸式地增长。在科学研究方面,随着研究手段的进步,数据量更是达到了前所未有的程度,例如在高能物理研究方面,很多实验每秒钟都能产生 TB 级别的数据量,但因为处理能力和存储能力的不足,科学家不得不把绝大部分未经处理的数据舍弃。无论是三维图形、海量数据处理、机器学习、语音识别,都需要极大的计算量。

算法并不局限于计算机和网络。在网络时代,越来越多的挑战需要靠卓越的算法来解决。如果你把计算机的发展放到数据飞速增长的大环境下考虑,你一定会发现,算法的重要性不是在日益减小,而是在日益增强。

在实际工程中我们遇到许多的高难度计算问题,有的问题在巨型计算机上采用一个劣质的算法来求解可能要数个月的时间,而且很难找到精确解。但采用一个优秀的算法,即使在普通的个人计算机上,可能只需数秒钟就可以求得解答。计算机求解一个工程问题的计

算速度不仅仅与计算机的设备水平有关,更取决于求解该问题的算法设计水平的高低。世界上许多国家,从大学到研究机关都高度重视对计算机算法的研究,已将提高算法设计水平看作是一个提升国家技术竞争力的战略问题。

对同一个计算问题,不同的人会有不同的计算方法,而不同算法的计算效率、求解精度和对计算资源的需求有很大的差别。

本书具体介绍枚举、递推、递归、回溯、动态规划、贪心算法与模拟等常用算法及其在实际案例求解中的应用。最后,介绍几个算法综合应用的案例。

1.1.2 算法描述

要使计算机能完成人们预定的工作,首先必须为如何完成这些工作设计一个算法,然后再根据算法编写程序。

一个问题可以设计不同的算法来求解,同一个算法可以采用不同的形式来表述。

算法是问题的程序化解决方案。描述算法可以有多种方式,如自然语言方式、流程图方式、伪代码方式、计算机语言表示方式与表格方式等。

当一个算法使用计算机程序设计语言描述时,就是程序。

本书采用 C 语言与自然语言相结合来描述算法。之所以采用 C 语言来描述算法,因为 C 语言功能丰富、表达能力强、使用灵活方便、应用面广,既能描述算法所处理的数据结构,又能描述计算过程,是目前大学阶段学习计算机程序设计的首选语言。

为方便算法描述与程序设计,下面把 C 语言的语法要点作简要概括。

1. 标识符

标识符可由字母、数字和下划线组成,但是标识符必须以字母或下划线开头。一个字母的大小写分别认为是两个不同的字符。

2. 常量

常量有以下几种类型。

- 整型常量:如十进制常数、八进制常数(以 0 开头的数字序列)、十六进制常数(以 0x 开头的数字序列)。
- 长整型常数(在数字后加字符 L 或 l);
- 实型常量(浮点型常量),具有小数形式与指数形式。
- 字符常量:用单引号(撇号)括起来的一个字符,可以使用转义字符;
- 字符串常量:用双引号括起来的字符序列。

3. 表达式

(1) 算术表达式

算术表达式有以下两种类型。

- 整型表达式:参加运算的运算量是整型量,结果也是整型数;
- 实型表达式:参加运算的运算量是实型量,运算过程中先转换成 double 型,结果为 double 型。

(2) 逻辑表达式

用逻辑运算符连接的整型量,结果为一个整数(0 或 1),逻辑表达式可以认为是整型表

达式的一种特殊形式。

(3) 字位表达式

用位运算符连接的整型量,结果为整数。字位表达式也可以认为是整型表达式的一种特殊形式。

(4) 强制类型转换表达式

用“(类型)”运算符使表达式的类型进行强制转换,如(float)a。

(5) 逗号表达式(顺序表达式)

形式如下:

表达式 1,表达式 2,...,表达式 n

顺序求出表达式 1,表达式 2,...,表达式 n 的值,结果为表达式 n 的值。

(6) 赋值表达式

将赋值号“=”右侧表达式的值赋给赋值号左边的变量,赋值表达式的值为执行赋值后被赋值的变量的值。注意,赋值号左边必须是变量,而不能是表达式。

(7) 条件表达式

形式如下:

逻辑表达式? 表达式 1: 表达式 2

逻辑表达式的值若为非 0(真),则条件表达式的值等于表达式 1 的值;若逻辑表达式的值为 0(假),则条件表达式的值等于表达式 2 的值。

(8) 指针表达式

对指针类型的数据进行运算。例如 p-2、p1-p2、&a 等(其中 p、p1、p2 均已定义为指针变量),结果为指针类型。

以上各种表达式可以包含有关的运算符,也可以是不包含任何运算符的初等量。例如,常数是算术表达式的最简单的形式。

表达式后加“;”,即为表达式语句。

4. 数据定义

对程序中用到的所有变量都需要进行定义,对数据要定义其数据类型,需要时要指定其存储类别。

(1) 数据类型标识符

数据类型标识符有 int(整型)、short(短整型)、long(长整型)、unsigned(无符号型)、char(字符型)、float(单精度实型)、double(双精度实型)、struct(结构体名)、union(共用体名)。

(2) 存储类别

存储类别有 auto(自动的)、static(静态的)、register(寄存器的)、extern(外部的)。

变量定义形式如下:

存储类别 数据类型 变量表列

例如:

```
static float x,y
```

5. 函数定义

存储类别 数据类型 <函数名>(形参表列)

{ 函数体 }

6. 分支结构

(1) 单分支

格式:

```
if(表达式) <语句 1>[ else <语句 2>]
```

功能: 如果表达式的值为非 0(真), 则执行语句 1; 否则(为 0, 即假)执行语句 2。所列语句可以是单个语句, 也可以是用 {} 界定的若干个语句。应用 if 嵌套可实现多分支。

(2) 多分支

格式:

```
switch(表达式)
{
    case 常量表达式 1: <语句 1>
    case 常量表达式 2: <语句 2>
    :
    case 常量表达式 n: <语句 n>
    default: <语句 n+1>
}
```

功能: 取表达式 1 时, 执行语句 1; 取表达式 2 时, 执行语句 2……其他所有情形, 执行语句 n+1。

case 常量表达式的值必须互不相同。

7. 循环结构

(1) while 循环

```
while(表达式) <语句>
```

功能: 表达式的值为非 0(条件为真), 执行指定语句(可以是复合语句)。直至表达式的值为 0(假)时, 脱离循环。

特点: 先判断, 后执行。

(2) do-while 循环

```
do <语句>
while(表达式);
```

功能: 执行指定语句, 判断表达式的值非 0(真), 再执行语句; 直到表达式的值为 0(假)时, 脱离循环。

特点: 先执行, 后判断。

(3) for 循环:

for(表达式 1;表达式 2;表达式 3)<语句>

功能: 解表达式 1;求表达式 2 的值: 若非 0(真),则执行语句;求表达式 3;再求表达式 2 的值……直至表达式 2 的值为 0(假)时,脱离循环。

以上三种循环,若执行到 break 语句,提前终止循环。若执行到 continue,结束本次循环,跳转下一次循环判定。

顺便指出,在不致引起误解的前提下,有时对描述的 C 语句进行适当简写或配合汉字标注,用以简化算法框架描述。

例如,用键盘输入整数 n ,按 C 语言的键盘输入函数应写为:

```
scanf("%d",&n);
```

描述时可简写为: input(n);

输入整数 n ,包括下面的输出描述。

要输出整数量 $a(1),a(2),\cdots,a(n)$,按 C 语言的输出函数应写为:

```
for(k=1;k<=n;k++)  
    printf("%d ",a[k]);
```

描述时可简写为: print($a(1)-a(n)$);

或简写为: 输出 $a(1:n)$;

例 1-1 求两个整数 $a,b(a>b)$ 的最大公约数的欧几里德算法:

(1) a 除以 b 得余数 r ;若 $r=0$,则 b 为所求的最大公约数。

(2) 若 $r\neq 0$,以 b 为 a , r 为 b ,继续(1)。

注意到任两个整数总存在最大公约数,上述辗转相除过程中余数逐步变小,相除过程总会结束。

解: 欧几里德算法又称为“辗转相除”法,算法具体描述如下:

```
input(a,b);  
if(a<b)  
    { c=a;a=b;b=c; }           //交换 a,b,确保 a>b  
r=a%b;  
while(r!=0)  
    { a=b;b=r;                 //实施“辗转相除”  
      r=a%b;  
    }  
print(b);
```

例 1-2 由 n 个 1 组成的整数能被 2011 整除, n 至少为多大?

解: 求解 n 至少为多大,应该从何下手?

下面使用模拟整数竖式除法。

被除数是 n 个 1,除数是 2011,竖式运算如下: