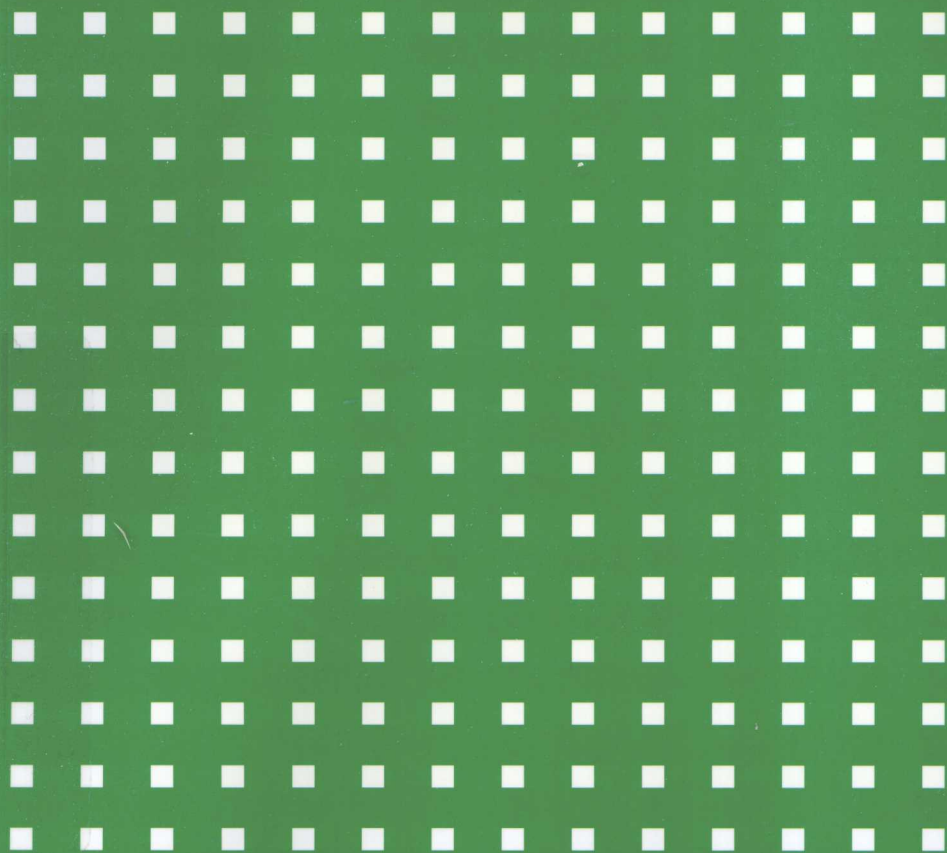


高等学校计算机专业教材精选·算法与程序设计

算法与程序设计 基础教程

冯俊 编著



清华大学出版社



内 容 简 介

本书在脱离具体语言环境下,围绕程序设计的中心问题——如何分析解决实际问题、如何构思算法,简明扼要地介绍了程序设计中的基本概念、基本思想、基本技术和基本方法。主要内容包括基本数据类型、构造数据类型和数据结构,基本控制结构、结构化程序设计技术和算法设计中的常用方法等。数据的组织形式采用具有丰富数据类型与良好结构的类 Pascal 语言进行描述,算法采用结构化流程图(N-S图)进行描述,所有算法以较流行的 C 语言编程实现。每一章都开辟了一个课程设计题目和习题,旨在提高读者解决问题的能力,同时也适合于自学。

本书条理清楚,内容翔实,概念表述严谨,逻辑推理严密,语言精练,用词达意,既注重程序设计思想介绍,又重视算法设计能力培养。算法构思精巧,结构清晰。本书深入浅出,配有大量实例和图示。

本书可作为计算机类专业或电子信息类专业或信息管理类相关专业的本专科教材,也可以作为使用计算机的广大科技工作者和管理工作者的参考用书。

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

算法与程序设计基础教程 / 冯俊编著. —北京:清华大学出版社,2010.2

(高等学校计算机专业教材精选·算法与程序设计)

ISBN 978-7-302-21361-1

I. ①算… II. ①冯… III. ①程序设计—高等学校—教材 ②电子计算机—算法理论—高等学校—教材 IV. ①TP3

中国版本图书馆 CIP 数据核字(2009)第 195557 号

责任编辑:汪汉友 白立军

责任校对:时翠兰

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 刷 者:北京四季青印刷厂

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185×260

印 张:19.75

字 数:476 千字

版 次:2010 年 2 月第 1 版

印 次:2010 年 2 月第 1 次印刷

印 数:1~4000

定 价:29.00 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:032915-01

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,并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了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教师,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,并力图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
2009年8月

前 言

当今世界已进入信息化时代,随着计算机科学技术的迅猛发展,计算机已对人类生活产生了巨大改变,愈来愈多的人开始学习使用计算机。程序设计是实现计算机应用的重要基础和有效手段。

20世纪80年代,计算技术开始渗透到大多数学科领域。1985年春,(美国)计算机协会(Association for Computing Machinery, ACM)与(美国)电气和电子工程师学会计算机分会(Institute of Electrical and Electronics Engineers-Computer Society, IEEE-CS)联手组成攻关组,经过近4年工作,提交了《计算作为一门学科》(Computing as a Discipline)的报告。1990年,攻关组在该报告的基础上又提交了计算学科教学计划的 Computing Curricula 1991(简称 CC1991)报告。1998年秋,ACM与IEEE-CS再次联手组成任务组,经过3年多的努力工作,于2001年12月提交了计算学科教学计划的 Computing Curricula 2001(简称 CC2001)报告。该报告将计算学科划分为14个领域,每个领域分成若干知识单元,每个知识单元又包括若干主题。CC2001将这些领域的大多数知识单元规定为计算机及相关学科本科学生必须掌握的核心知识。程序设计基础主要涉及程序设计基础(PF)、算法与复杂性(AL)和程序设计语言(PL)等领域。掌握这些领域的知识对于利用计算机资源,开发高效的程序系统是必不可少的。本书内容涵盖或涉及上述3个领域中的多个知识单元,主要包括PF1(基本程序设计结构)、PF2(算法和问题求解)、PF3(基本数据结构)、PF4(递归技术)、PF5(事件驱动程序设计)、AL1(基本算法分析)、AL2(算法策略)、AL3(基本计算算法)、PL1(程序设计语言概述)、PL3(语言翻译)、PL4(数据类型)、PL6(面向对象程序设计)和PL9(数据抽象)等。

计算机运行的过程就是程序执行的过程,运用计算机解决现实世界中的任何实际问题,最终都要将现实问题转换成计算机程序,在这一转换过程中,需要运用多方面的知识进行程序设计,程序是程序设计的结晶,程序设计是开发和应用计算机的钥匙。

“程序设计基础”课程是为高校大多数专业设置的,长期以来,关于如何讲授程序设计基础课程,许多人把争论的焦点放在了语言的选择上,把讲授的重点放在了语言本身,却忽略了程序设计真正实用的基本思维方式和方法,最后导致学生几乎没有分析问题、解决问题的技能。程序设计基础课程教学的核心目标,应该是让学生学习和掌握对于实际问题,如何分析问题和设计解决它们的算法;帮助学生理解程序设计的基本思想和科学原理;掌握程序设计的基本知识、基本技术和基本方法;掌握程序设计中的数据组织结构和程序流程控制结构。为学生能用计算机处理实际问题打下良好基础。

本书主要内容大体可分为五个组成部分。第1章为第一部分,主要介绍程序设计的基础知识。包括程序设计的基本概念、基本思想、基本方法和基本步骤等。帮助读者理解这些概念,弄清这些术语之间的关系,为学习后续内容打好基础。第2、4、6章为第二部分,主要阐述程序设计中的数据组织。包括基本数据类型、构造数据类型和基本数据结构。帮助读者掌握数据的组织形式,能够选择合适的数据组织形式解决实际问题。第3章与第5章的

5.6、5.7节为第三部分,主要阐述程序设计中的控制结构。包括三种基本控制结构、函数、子程序与过程文件等内容。帮助读者掌握程序的三种基本控制结构(顺序结构、选择结构、循环结构)与实现过程的封装。在解决实际问题中,能够设计出具有良好结构的程序。第5章的5.1~5.5节与第7章为第四部分,主要阐述程序设计中的技术与方法。包括程序设计的技术与方法(模块化设计技术与方法、自顶向下设计技术与方法、逐步求精设计技术与方法、结构程序优化技术与方法等)以及算法设计中的常用方法(包括枚举法、递归与递推、分治法、动态规划法、贪心法、回溯法与分支限界法等),在这一部分将通过能付诸实施的、生动有趣的大量应用实例进行讲解,使读者进一步熟悉前面所学内容,熟练掌握程序设计的基本思想、基本技术和基本方法。第8章为第五部分,以解决问题为中心。通过几个典型问题,综合运用前面所学知识,完成问题分析、算法设计、程序实现与程序运行。使所学知识综合应用到解决实际问题中。

由于 Pascal 语言具有丰富的数据类型和良好的结构,所以在数据组织的描述中,拟选用类 Pascal 语言作为工具;为了着重体现算法设计的思想与算法结构,对算法的描述拟选用结构化流程图(N-S图)作为工具;为了方便读者上机实践,将选用较流行的 C 语言或 VFP 语言对所有算法进行编程实现。既让读者在脱离复杂语言环境下轻松学习程序设计思想和方法,又不至于使他们陷入只有思想的纸上谈兵。这是本书的特色之一。程序设计既是一门实践性很强的带有艺术特性的变换技术,又是一门科学。本书在每一章都开辟了一个课程设计题目,包括问题描述、基本要求、测试数据、实现提示和问题拓展,旨在提高读者分析问题和解决问题的能力。这是本书的又一个特色。

程序设计基础课程在教学计划中至少应为 6 学分,课堂教学在 60~80 学时之间。本课程是一门技术性、实践性都很强的课程,为了使学能真正掌握有关理论知识和应用技术,在整个教学过程中至少应安排 5 个以上课程设计,必须保证学生有足够的课下思考作业时间和上机实践时间。上机时数、课下作业时数和课堂讲授时数的比例应不低于 0.5:2:1。

本书作为我校教改项目——“数据结构与程序设计课程建设及教学现代化研究”的组成部分,在课程建设过程中,得到各级领导的支持、专家的指导和同事的帮助。在此表示谢意。

本书凝结了作者二十多年来的教学科研成果和在讲授“程序设计基础”、“数据结构”等课程中的教学经验。在编写过程中,参阅了多种程序设计基础与算法的优秀教材。在编辑出版过程中,得到了清华大学出版社各级领导的支持;负责本书编辑工作的全体同仁,特别是责任编辑汪汉友同志付出了辛勤劳动。在此一并表示衷心的感谢。

由于作者水平有限,加之学科理论与技术发展日新月异,书中疏漏谬误之处在所难免,恳请广大读者指正。E-mail: fengj1682000@tom.com。

作 者

2009年8月

目 录

第 1 章 绪论	1
1.1 程序=数据结构+算法	1
1.1.1 程序.....	1
1.1.2 什么是数据结构.....	1
1.1.3 什么是算法.....	5
1.2 程序设计=数据结构+算法+程序设计方法	6
1.2.1 程序设计.....	7
1.2.2 程序设计方法学.....	7
1.3 语言工具和环境.....	11
1.3.1 程序设计语言	11
1.3.2 程序设计范型	13
1.3.3 程序设计语言的语法元素及其元素功能	14
1.3.4 编程环境和程序运行	15
1.4 程序设计步骤与程序设计风格.....	17
1.4.1 程序设计步骤	17
1.4.2 程序设计风格	18
1.5 算法设计与分析.....	19
1.5.1 算法描述	19
1.5.2 算法设计举例	22
1.5.3 算法设计的要求	26
1.5.4 算法分析	26
1.6 课程设计相关知识.....	28
1.6.1 课程设计目的与内涵	28
1.6.2 课程设计步骤	28
1.6.3 课程设计报告规范	29
1.7 课程设计题目——求最大公因子.....	30
习题	30
第 2 章 简单数据类型与表达式	32
2.1 数据类型.....	32
2.1.1 基本概念和术语	32
2.1.2 数据类型与数据结构	33
2.1.3 简单数据类型	33
2.1.4 构造数据类型	34

2.2	常量与变量	35
2.2.1	常量	35
2.2.2	变量	37
2.3	运算符与表达式	39
2.3.1	算术运算符与算术表达式	39
2.3.2	字符运算符与字符表达式	41
2.3.3	关系运算符与关系表达式	41
2.3.4	逻辑运算符与逻辑表达式	42
2.4	课程设计题目——求最小公倍数	45
	习题	45
第3章	程序的基本控制结构	47
3.1	程序的基本控制结构介绍	47
3.1.1	3种基本控制结构	47
3.1.2	关于对GOTO语句的认识	49
3.2	顺序结构程序设计	50
3.3	选择结构程序设计	53
3.3.1	单向分支选择结构程序设计	54
3.3.2	双向分支选择结构程序设计	56
3.3.3	多向分支选择结构程序设计	60
3.4	循环结构程序设计	62
3.4.1	当型循环结构程序设计	63
3.4.2	直到型循环结构程序设计	66
3.4.3	步长型循环结构程序设计	68
3.5	课程设计题目——求解方程的根	73
	习题	74
第4章	构造数据类型	76
4.1	数组类型	76
4.1.1	一维数组	76
4.1.2	二维数组	79
4.1.3	查找	83
4.1.4	排序	87
4.2	结构体类型	92
4.2.1	结构体类型的概念	92
4.2.2	结构体类型的定义	92
4.2.3	结构体变量的说明	93
4.2.4	结构体变量的引用	94
4.2.5	结构体应用举例	95

4.3	其他构造数据类型	102
4.3.1	共用体类型	102
4.3.2	文件类型	105
4.4	抽象数据类型	112
4.5	课程设计题目——排序算法	113
	习题	114
第5章	结构化程序设计	116
5.1	结构化方法概述	116
5.2	模块化设计技术与方法	117
5.2.1	模块化的一般目标	117
5.2.2	模块凝聚(聚合)与模块耦合(关联)	117
5.2.3	模块的设计准则	118
5.3	自顶向下设计技术与方法	120
5.3.1	自顶向下设计	121
5.3.2	自顶向下编码	121
5.4	逐步求精设计技术与方法	122
5.4.1	选择排序算法的逐步求精设计过程	122
5.4.2	积木游戏算法的逐步求精设计过程	125
5.5	结构程序优化技术和方法	129
5.5.1	问题模型的优化	129
5.5.2	计算方法的优化	130
5.5.3	算法的优化	132
5.5.4	数据结构的优化	132
5.6	子程序与过程文件	133
5.6.1	子程序	133
5.6.2	过程文件	133
5.6.3	过程应用举例	134
5.7	函数	139
5.7.1	函数的定义和调用	139
5.7.2	函数的嵌套调用和递归调用	141
5.7.3	函数应用举例	144
5.8	课程设计题目——学生成绩管理系统	149
	习题	150
第6章	基本数据结构	151
6.1	顺序表	151
6.1.1	向量的顺序存储表示	152
6.1.2	向量的运算	152

6.1.3	应用举例	158
6.2	链表	160
6.2.1	指针与指针对象	161
6.2.2	单链表	163
6.2.3	应用举例	171
6.3	栈	174
6.3.1	栈的概念	174
6.3.2	顺序栈	175
6.3.3	链接栈	178
6.4	递归与非递归过程	180
6.4.1	递归的概念	180
6.4.2	递归过程(函数)设计	182
6.4.3	递归过程与非递归过程	183
6.5	队列	187
6.5.1	队列的概念	187
6.5.2	顺序队列	187
6.5.3	链接队列	190
6.6	二叉树	194
6.6.1	树的基本概念	194
6.6.2	二叉树介绍	195
6.6.3	二叉树的存储表示	197
6.6.4	二叉树的遍历	198
6.7	课程设计题目——一元多项式计算器	209
	习题	210
第7章	算法设计中的常用方法	213
7.1	问题的解空间	213
7.2	枚举法	214
7.2.1	枚举法的基本思想	215
7.2.2	枚举法应用举例	215
7.2.3	枚举算法的优化	217
7.3	递归与递推	223
7.3.1	梵天塔问题	223
7.3.2	再谈递归算法设计	224
7.3.3	快速排序	225
7.3.4	递推算法	229
7.3.5	Wythoff 数对序列	232

7.4	分治法	235
7.4.1	分治法概述	235
7.4.2	数字旋转方阵	236
7.4.3	最大子段和问题	238
7.5	动态规划法	242
7.5.1	动态规划法概述	242
7.5.2	多段图的最短路径问题	243
7.5.3	0-1 背包问题	246
7.6	贪心法	249
7.6.1	贪心法概述	249
7.6.2	背包问题	250
7.6.3	0-1 背包问题及贪心 k 阶优化方法	252
7.7	回溯法	254
7.7.1	回溯法概述	254
7.7.2	0-1 背包问题与回溯递归算法	255
7.7.3	0-1 背包问题与回溯迭代算法	258
7.8	分支限界法	261
7.8.1	分支限界法概述	261
7.8.2	分支限界法求解 0-1 背包问题	263
7.9	课程设计题目——0-1 背包问题	267
	习题	268
第 8 章	以解决问题为中心	270
8.1	一元多项式问题	270
8.1.1	问题描述	270
8.1.2	问题分析	270
8.1.3	算法设计	272
8.1.4	C 语言程序实现与程序运行	274
8.2	八皇后问题	277
8.2.1	问题描述	277
8.2.2	问题分析	277
8.2.3	算法设计	277
8.2.4	C 语言程序实现与程序运行	282
8.2.5	VFP 语言程序实现与程序运行	284
8.3	骑士游历问题	286
8.3.1	问题描述	286
8.3.2	问题分析与算法设计	286

8.3.3 C 语言程序实现与程序运行	288
8.4 哈夫曼树与哈夫曼编码	290
8.4.1 问题描述	290
8.4.2 问题分析与算法设计	292
8.4.3 C 语言程序实现与程序运行	295
8.5 课程设计题目——哈夫曼编/译码系统	298
习题	299
参考文献	300

第 1 章 绪 论

程序和程序设计是计算学科中最基本、最重要的概念。计算机运行的过程就是程序执行的过程,运用计算机解决现实世界中的任何实际问题,最终都要将现实问题转换成计算机程序,在这一转换过程中,需要运用多方面的知识进行程序设计,程序是程序设计的结晶,程序设计是开发和应用计算机的钥匙。本章将介绍程序设计的有关基本概念及其基础知识。

1.1 程序=数据结构+算法

数据结构与算法是计算学科中研究的基本课题。世界著名的计算机科学家、Pascal 语言的发明者、第 19 位图灵奖(1984 年)获得者 N. 沃思(Niklaus Wirth)教授曾提出了这样一个有名的公式:

$$\text{程序} = \text{数据结构} + \text{算法}$$

它清楚地揭示了计算机科学中数据结构与算法这两个概念的重要性和统一性。人们不能离开数据结构去抽象地分析求解问题的算法,也不能脱离算法去孤立地研究程序的数据结构。N. 沃思教授还说,不了解施加于数据上的算法,就无法决定如何构造和组织数据;反之,算法的选择也常常在很大程度上要依赖于作为基础的数据结构。

1.1.1 程序

“程序”一词,从广义上讲可以认为是一种行动方案或工作步骤。这里的程序指的是计算机程序(Program),它实际上表示的也是一种处理事务的步骤和顺序。由于组成计算机程序的基本单位是指令,因此,计算机程序就是按照操作步骤事先编制好的、具有特定功能的有限指令序列。

一个计算机程序必须对问题的每个对象和处理规则给出正确详尽的描述。针对问题所要处理的对象,设计合理的数据结构,常常可以有效地简化算法,数据结构与算法是计算机程序的两个重要方面。数据结构是加工处理的对象,一个计算机程序要进行计算或处理总是以某些数据元素为对象,要设计一个好的程序就需要将这些数据按照某种要求组织成一个适合的数据结构。算法是程序的核心,它在程序编制、软件开发,乃至在整个计算机科学中都占有重要地位。程序是算法和数据结构两要素统一的全过程,或者说,程序就是在数据的某种特定表示方式以及结构的基础上对抽象算法用某种程序设计语言进行的具体描述(实现)。程序与算法不同,程序不一定满足算法要遵循的准则。例如,操作系统是一个在无限循环中执行的程序,它不是一个算法。

1.1.2 什么是数据结构

数据结构是随着计算机科学技术的发展而逐渐形成的一门学科,它是计算机相关专业的核心课程。当今,计算机应用已渗透到人类社会的各个领域,除了用于科学计算之外,更

广泛地用于科学管理等方面。因此,计算机处理的数据量越来越大,数据间的关系也越来越复杂,这就要求人们必须研究如何有效地组织数据和处理数据,这正是数据结构要研究的内容。下面通过一个例子说明数据结构在计算机科学技术中的重要地位。

例 1-1 一个工厂生产模型。

工厂的生产过程可以看成是对原材料的加工处理,最后得出产品的过程。在这个过程中,显然包括两个关键阶段:

- (1) 原材料的管理——原材料如何在仓库中进行组织、存储和管理。
- (2) 原材料的加工处理——即采用什么样的工艺技术、按照什么样的操作顺序对原材料进行加工处理,最后得到合格产品。

工厂生产模型如图 1-1 所示。

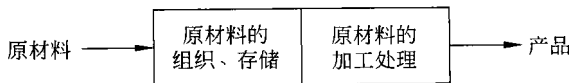


图 1-1 工厂生产模型

由此可见,原材料的组织、存储管理和原材料的加工处理是工厂进行正常生产的关键。

学习计算机科学的目的是运用计算机来解决实际问题。计算机的解题过程也可以看作是对原材料进行组织、存储管理、加工处理,最后得到产品的过程。只不过这里的原材料是数据,产品是处理结果,对数据的加工处理是由算法决定的。

大家知道:数据是对客观事物采用计算机能够识别、存储和处理的形式所进行的描述。随着计算机的发展和应用领域的扩大,数据量越来越大,数据间的联系也越来越复杂。因此,对数据组织结构的研究也就越来越受到重视。

计算机解题模型如图 1-2 所示。

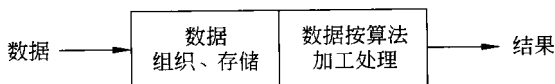


图 1-2 计算机解题模型

由此可见,计算机解题的关键是数据的组织和算法的设计。数据结构就是研究数据组织和算法设计的。

简单地说,数据结构(Data Structure)是研究一些数据的集合。就是根据数据的性质、数据元素之间的关系,研究如何表示、存储、操作这些数据的技术。实现数据结构的存储和操作都涉及算法,它们之间存在着本质的联系,失去一方,另一方就没有什么意义。如:当讨论一种类型的数据结构时,总离不开讨论对这种类型数据结构需要施加的各种运算,只有通过对这些运算的算法进行研究,才能更清楚地理解这种数据结构的意义和作用;反过来,当讨论一个算法时,也总是自然地联系到作为该算法的处理对象——数据的组织及其存储表示。

在计算学科教学计划 2001(Computing Curricula 2001,CC2001)的 14 个研究领域,数据结构和算法的基本内容主要涵盖在程序设计基础(Programming Fundamentals,PF)、算法与复杂性(Algorithms and Complexity,AL)和程序设计语言(Programming Language,

PL)3个领域中。CC2001强调了算法和程序设计。由此可见,人们越来越重视数据结构和算法,认为程序设计的实质就是对确定的问题,选择一种好的数据结构和设计一个好的算法。因此,“数据结构”课程已不仅是计算机相关专业教学计划中的核心课程之一,而且也是非计算机专业的主要选修课程之一。

究竟什么是数据结构?当使用计算机来解决实际问题时,一般需要经过这样几个步骤:首先要从问题中抽象出一个适当的数学模型,然后设计一个解决数学模型的算法,最后编出程序、调试程序、进行测试直至得到最终的解答。对于求解数值计算问题,数学模型一般可以用数学方程加以描述。然而,对于更多的求解非数值计算问题,通常是无法用数学方程来描述的,这类问题数据量大、数据间的关系复杂,求解的不是某个数值或几个数值,而是要得到某种检索结果、某种排列状态或某种设计的表示等,这类问题通常是用一种被称为数据结构的工具来描述数据及其数据之间的关系。下面通过一个例子来认识数据结构。

例 1-2 图书馆的书目检索问题。

当你想借阅一本参考书又不知道书库中是否有此书的时候,或者当你想找某一方面的参考书又不知道图书馆内有哪些这方面书的时候,都需要到图书馆去查阅图书目录卡片。在图书馆内有各种名目的卡片:有按书名编排的,有按作者编排的,还有按分类编排的等。若利用计算机进行检索,则计算机处理的对象就是这些目录卡片上的书目信息。列在卡片上的一本书的书目信息由登录号、书名、作者名、分类号、出版单位和出版时间等若干数据项组成,每一本书都有一个唯一的登录号,但是不同的书目之间可能有相同的书名、或者有相同的作者名、或者相同的分类号。因此,在书目自动检索系统中,可以建立一个按登录号顺序排列的书目文件和三个分别按书名、作者名和分类号顺序排列的索引表,如图 1-3 所示。这四张表就是书目检索问题的数学模型描述。

登录号	书名	作者	分类号	...
1001	高等数学	樊映川	S01	...
1002	理论力学	罗远祥	L01	...
1003	高等数学	华罗庚	S01	...
1004	线性代数	溱汝书	S02	...
...

(a) 按登录号排列

书名	登录号
高等数学	1001,1003
理论力学	1002, ...
线性代数	1004, ...
...	...

(b) 按书名排列

作者	登录号
樊映川	1001, ...
罗远祥	1002, ...
华罗庚	1003, ...
溱汝书	1004, ...

(c) 按作者名排列

分类	登录号
S	1001,1003,
L	1002, ...
...	...

(d) 按分类号排列

图 1-3 书目文件和索引表的示例

这个例子中的问题是非数值计算问题,其数学模型无法用数学公式或数学方程来描述,它使用了被称为表的数据结构来加以描述。或者简单地说,数据结构是指数据及其数据之间的关系。对于数据结构这个概念,至今尚未有一个被一致公认的标准定义。不过在讨论任何一种数据结构时,都会自然地联系到这种类型的数据所需要的运算,以及为了在计算机

上实现这些运算,如何将这些数据存储到计算机中。因此,在讨论数据结构的时候,一般考虑以下三方面。

(1) 数据之间的逻辑关系——数据的逻辑结构。

(2) 数据在计算机中的存储形式——数据的存储结构。

(3) 定义在逻辑结构上的运算及其在存储结构上的运算的算法实现——数据的运算。

例如,对于一个给定的线性表。它的逻辑结构是指:哪个元素是表中的第一个元素;哪个元素是表中的最后一个元素;哪个元素是在给定元素之前或之后等。它的存储结构是指:元素在存储器中是顺序地邻接存放,还是链接存储等。它的运算是指:在表中查找一个元素;从表中删除一个元素;在表中插入一个元素等。但是,不能孤立地去理解某个方面,它们之间存在着内在的密切联系。

例 1-3 设有某单位职工工资表,如表 1-1 所示。

表 1-1 职工工资表

编号	姓名	基本工资	津贴	扣除费	实发工资
1001	王一华	880	400	200	1080
1002	李静	840	360	180	1020
1003	张丽华	930	500	210	1220
...

这张工资表可以看作是一个数据结构。表中的每一行反映的是一位职工的工资信息,把它看作是一个整体,称为一个**结点**或**数据元素**,它是数据结构中讨论的基本单位,即数据结构是结点的集合。这张工资表也可以看作是一个线性表,因为,表中结点与结点之间的关系是一种简单的线性关系,表中有且仅有一个结点是表头结点,有且仅有一个结点是表尾结点,表中的任一结点,与它相邻且在它前面或后面的结点最多只有一个,这就是它的逻辑结构。当将这个线性表存入计算机时,是采用顺序存储方式还是采用链接存储方式,这是它的存储结构要解决的问题。至于数据的运算问题,指的是对表中的结点进行查找、修改、删除、插入等操作。这些问题都弄明白了,该表的数据结构也就完全清楚了。

1. 数据的逻辑结构

数据的逻辑结构(Logical Structure)是对数据间关系的描述,形式地可以用一个二元组表示:

$$DS = (K, R)$$

其中:K 是结点的有穷集合,R 是 K 上的关系的有穷集合,每个关系都是从 K 到 K 上的结点序偶的集合。在不易产生混淆的情况下,有时也把数据的逻辑结构简称为数据结构。

2. 数据的存储结构

数据的逻辑结构是从逻辑关系的角度来考察数据的,它是面向问题实体的,是独立于计算机的。而数据的存储结构(Storage Structure)是逻辑结构在计算机存储器里的实现,它是依赖于计算机的。研究数据的存储结构,一方面要考虑使逻辑结构组织好的数据完整地存储到计算机的存储器中;另一方面还要考虑使运算能够较好地实现。

因此,对数据的逻辑结构 $DS=(K, R)$ 的存储表示(数据的存储结构)需要考虑以下三方面问题:

- (1) 保证结点集合 K 中的每个结点 k 都存储到计算机的存储器中。
- (2) 这种存储方式必须明显或隐含地体现结点间的联系,即关系 R 。
- (3) 这种存储方式应该能使定义在逻辑结构上的一组运算较好地实现。

在实际应用中,数据的存储方法是灵活多样的,可根据问题规模和运算种类等因素适当选择。基本存储方法有四种:顺序存储方法、链接存储方法、索引存储方法和散列存储方法。

3. 数据的运算

研究数据结构是为了更有效地处理数据和使用数据。对数据的处理一般称为数据的运算(Operation)。数据的运算是定义在数据的逻辑结构之上的,运算的具体实现是在数据的存储结构上进行的。执行运算可以对数据结构中的数据元素实施相应的操作。

在不同的应用场合,对数据会有各种各样的运算要求。亦即每种逻辑结构都有一个运算的集合。最常用的运算有以下几种:

- (1) **检索(Search)**: 又称为查找,就是在数据结构中查找满足一定条件的结点。
- (2) **插入(Insert)**: 在数据结构中的指定位置插入一个新结点。
- (3) **删除>Delete): 将数据结构中指定的结点删除。**
- (4) **更新(Update)**: 修改数据结构中指定结点的的一个或多个字段的值。

插入、删除、更新运算都包含着一个检索运算。

(5) **排序(Sort)**: 保持结构的结点集合中结点数不变,将结点按照某种指定的顺序重新排列。

数据的运算是数据结构三个方面中不可分割的一个重要方面,有时由于运算性质的不同,将会导致数据结构的很大差别。例如,对于一个线性表,若所有的插入、删除都限制在表的一端进行,则该表称为栈。若所有的插入限定在表的一端进行,所有的删除限定在表的另一端进行,则该表称为队列。

研究数据结构是为了解决实际应用问题,解决问题的实质是对数据进行有效地处理。因此,在讨论数据结构时,不但要讨论数据的逻辑结构,讨论数据的存储结构,还要讨论在逻辑结构上定义的运算集合以及在存储结构上实现这些运算的算法。通过对运算及其实现这些运算的算法的性能分析,使得在求解实际应用问题时,能有效地选择和设计适当的数据结构,编写出高效的程序。

综上所述,按照某种逻辑关系组织起来的一批数据,并在其上定义了一个运算的集合;按照一定的存储表示方式把它存储到计算机中,并在其上实现这一组运算,这就是一个数据结构。

在数据结构的论述中,人们所探讨的内容是:对于一个给定的问题,它的数据在逻辑上有什么特征;应该如何来组织它,才便于人们对数据进行处理和利用。

1.1.3 什么是算法

如前所述,数据结构与算法之间存在着密切的联系。可以说,不了解施加于数据上的算法需求,就无法决定数据结构;反之,算法的方法选择和结构设计在很大程度上又依赖于作为基础的数据结构。大家知道,在程序设计过程中,有相当多的时间是花费在考虑如何解决问题上,即通常所说的构思算法,一旦有了合适的算法,用某种程序设计语言来编写程序并