

可下载教学资料

<http://www.tup.tsinghua.edu.cn>

21世纪普通高校计算机公共课程规划教材

数据结构 (C语言版)

第3版

邓文华 主编 邹华胜 副主编



清华大学出版社

21世纪普通高校计算机公共课程规划教材

数据结构（C语言版）

第3版

邓文华 主编 邹华胜 副主编

清华大学出版社
北京

内 容 简 介

本书对常用的数据结构作了系统的介绍,力求概念清晰,注重实际应用。全书共分9章,依次介绍了数据结构的基本概念、线性表、栈和队列、串与数组、树与二叉树、图结构,以及查找和排序等基本运算,第9章给出了6个综合应用的实例程序。全书用C语言作为算法描述语言,每一章均附有典型例题与小结,便于总结与提高。

本书叙述简洁、深入浅出,主要面向独立学院三本计算机类专业的学生,也可以作为大学非计算机专业学生的选修课教材和计算机应用技术人员自学参考用书。

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

数据结构: C语言版/邓文华主编. —3版. —北京:清华大学出版社,2011.1

(21世纪普通高校计算机公共课程规划教材)

ISBN 978-7-302-23598-9

I. ①数… II. ①邓… III. ①数据结构—高等学校:技术学校—教材 ②C语言—程序设计—高等学校:技术学校—教材 IV. ①TP311.12 ②TP312

中国版本图书馆CIP数据核字(2010)第159189号

责任编辑:梁颖 顾冰

责任校对:焦丽丽

责任印制:杨艳

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,jsjjc@tup.tsinghua.edu.cn

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

印 刷 者:北京富博印刷有限公司

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:15 字 数:358千字

版 次:2011年1月第3版 印 次:2011年1月第1次印刷

印 数:1~3000

定 价:25.00元

产品编号:036979-01

出版说明

随着我国改革开放的进一步深化,高等教育也得到了快速发展。各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度。通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程(简称‘质量工程’)”。通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建议,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

本系列教材立足于计算机公共课程领域,以公共基础课为主、专业基础课为辅,横向满足高校多层次教学的需要。在规划过程中体现了如下一些基本原则和特点。

(1) 面向多层次、多学科专业,强调计算机在各专业中的应用。教材内容坚持基本理论适度,反映各层次对基本理论和原理的需求,同时加强实践和应用环节。

(2) 反映教学需要,促进教学发展。教材要适应多样化的教学需要,正确把握教学内容和课程体系的改革方向,在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养,为学生知识、能力、素质协调发展创造条件。

(3) 实施精品战略,突出重点,保证质量。规划教材把重点放在公共基础课和专业基础课的教材建设上;特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现教学质量和教学改革成果的教材。

(4) 主张一纲多本,合理配套。基础课和专业基础课教材配套,同一门课程有针对不同层次、面向不同专业的多本具有各自内容特点的教材。处理好教材统一性与多样化,基本教材与辅助教材、教学参考书,文字教材与软件教材的关系,实现教材系列资源配套。

(5) 依靠专家,择优选用。在制定教材规划时要依靠各课程专家在调查研究本课程教

材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主题。书稿完成后要认真实行审稿程序,确保出书质量。

繁荣教材出版事业,提高教材质量的关键是教师。建立一支高水平教材编写梯队才能保证教材的编写质量和建设力度,希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21世纪普通高校计算机公共课程规划教材编委会

联系人:梁颖 liangying@tup.tsinghua.edu.cn

前 言

作为独立学院三本的学生有着其自身的特点,如果直接拿原来一本、二本的教材给独立学院的三本学生使用是很不恰当的,本书是在保留原第 2 版精华基础上,专门针对独立学院三本的学生特点,即理论够用,着重培养学生设计能力与动手能力而编写的。

“数据结构”是计算机程序设计的重要理论基础,是计算机及其应用专业的一门重要基础课程和核心课程。它不仅是计算机软件专业课程的先导,而且也逐渐为其他工科类专业所重视。

本书共分 9 章,第 1 章叙述数据、数据结构和算法等基本概念。第 2~6 章分别讨论了线性表、栈和队列、串和数组、树与二叉树,以及图等基本数据结构的实现及其应用。第 7、8 章分别讨论了查找和排序操作的各种实现方法及其特点,第 9 章给出了 6 个综合应用的实例程序,以便学生上机实习,进一步加强对学生的实际应用能力及动手能力的培养。

本书具有以下特点:

(1) 基础理论知识的阐述由浅入深、通俗易懂。内容组织和编排以应用为主线,略去了一些理论推导和数学证明的过程,淡化算法的设计分析和复杂的时空分析。

(2) 各章都配有相应的典型例题或应用,列举分析了许多实用的例子,大多数算法都直接给出了其相应的 C 语言程序,以便学生上机练习、实践。

(3) 为了便于学生复习及掌握每章的重点、要点,本书在每章的结束处对该章进行小结,并列本章的重点。

(4) 本书配有电子教案及习题解答,方便教师的备课与教学。

本书讲课时数为 60~90 学时左右,上机时数在 20~36 学时左右。教师可根据学时数、专业和学生的实际情况选讲应用与典型例题中的例子。

本书由邓文华任主编,邹华胜任副主编。其中第 1、2 章由邹华胜编写,第 3、4 章由胡智文编写,第 5、6 章由毕保祥编写,第 7~9 章由邓文华编写,谢胜利也编写了第 7、8 章的部分内容。全书由邓文华统稿。

编写独立学院三本的教材是一项新的尝试,书中难免存在疏漏,恳请读者批评指正。

编 者

目 录

第 1 章 绪论	1
1.1 从问题到程序	1
1.2 有关概念和术语	4
1.3 算法及算法分析	6
1.3.1 算法特性.....	6
1.3.2 算法描述.....	7
1.3.3 算法分析.....	8
1.4 关于数据结构的学习.....	10
1.5 关于本书的编写说明.....	11
本章小结	12
习题 1	12
第 2 章 线性表	15
2.1 线性表的逻辑结构.....	15
2.1.1 线性表的定义	15
2.1.2 线性表的基本操作	15
2.2 线性表的顺序存储及其操作的实现.....	17
2.2.1 顺序表	17
2.2.2 顺序表基本操作的实现	18
2.2.3 顺序表的其他操作举例	21
2.3 线性表的链式存储及其操作的实现.....	23
2.3.1 单链表	23
2.3.2 单链表基本操作的实现	25
2.3.3 循环链表	30
2.3.4 双向链表	31
2.3.5 单链表的其他操作举例	33
2.4 典型应用.....	35
2.4.1 一元多项式的存储表示	35
2.4.2 一元多项式的相加运算	36
本章小结	37
习题 2	38

第3章 栈和队列	40
3.1 栈.....	40
3.1.1 栈的定义及其基本运算	40
3.1.2 栈的存储结构和基本运算的实现	41
3.1.3 栈的应用举例	43
3.1.4 栈与递归的实现	47
3.2 队列.....	52
3.2.1 队列的定义及其基本运算	52
3.2.2 队列的存储结构和基本运算的实现	52
3.2.3 队列的应用举例	57
3.3 典型例题.....	58
本章小结	60
习题3	61
第4章 串和数组	65
4.1 串.....	65
4.1.1 串的基本概念	65
4.1.2 串的基本运算	66
4.1.3 串的存储结构及其基本运算的实现	67
4.1.4 串的其他运算举例	69
4.2 数组.....	70
4.2.1 数组的逻辑结构和基本操作	70
4.2.2 数组的存储结构	71
4.2.3 稀疏矩阵	72
4.2.4 矩阵的其他运算举例	75
4.3 典型例题.....	76
本章小结	77
习题4	78
第5章 树与二叉树	80
5.1 树的概念与基本操作.....	80
5.1.1 树的定义及相关术语	80
5.1.2 树的基本操作	82
5.2 二叉树.....	82
5.2.1 二叉树的基本概念	82
5.2.2 二叉树的主要性质	84
5.2.3 二叉树的存储结构与基本操作	86
5.2.4 二叉树的遍历	89
5.2.5 二叉树的其他操作举例	93

5.3	树与森林	96
5.3.1	树的存储	96
5.3.2	树、森林与二叉树的相互转换	98
5.3.3	树和森林的遍历	101
5.4	最优二叉树——哈夫曼树	102
5.4.1	哈夫曼树的基本概念	102
5.4.2	哈夫曼树的构造算法	103
5.4.3	哈夫曼编写	105
5.4.4	哈夫曼编码的算法实现	106
5.5	典型例题	107
	本章小结	110
	习题 5	110
第 6 章	图	114
6.1	图的基本概念	114
6.1.1	图的定义和术语	114
6.1.2	图的基本操作	117
6.2	图的存储结构	117
6.2.1	邻接矩阵	117
6.2.2	邻接表	119
6.3	图的遍历	121
6.3.1	深度优先搜索	121
6.3.2	广度优先搜索	123
6.4	图的应用	124
6.4.1	最小生成树	124
6.4.2	最短路径	129
6.4.3	拓扑排序	131
6.5	典型例题	134
	本章小结	138
	习题 6	138
第 7 章	查找	142
7.1	基本概念与术语	142
7.2	静态查找表	143
7.2.1	静态查找表结构	143
7.2.2	顺序查找	144
7.2.3	有序表的折半查找	145
7.2.4	分块查找	148
7.3	动态查找表	148
7.4	哈希表	157



7.4.1	哈希表与哈希方法.....	157
7.4.2	常用的哈希函数构造方法.....	157
7.4.3	处理冲突的方法.....	159
7.4.4	哈希表的查找算法.....	161
7.4.5	哈希表的性能分析.....	162
7.5	典型例题	163
	本章小结.....	168
	习题7	168
第8章	排序	172
8.1	基本概念	172
8.2	三种简单排序方法	173
8.2.1	直接插入排序.....	173
8.2.2	冒泡排序.....	174
8.2.3	简单选择排序.....	177
8.3	希尔排序	178
8.4	快速排序	179
8.5	堆排序	182
8.6	归并排序	184
8.7	基数排序	186
8.7.1	多关键码排序.....	186
8.7.2	链式基数排序.....	187
8.8	各种排序方法的比较与讨论	189
8.9	典型例题	190
	本章小结.....	193
	习题8	193
第9章	综合应用实例	197
9.1	上机实验要求及规范	197
9.1.1	上机实习的具体步骤.....	197
9.1.2	实验报告的基本要求.....	198
9.2	约瑟夫环问题	199
9.3	迷宫问题	201
9.4	短信促销活动	206
9.5	保龄球记分系统	212
9.6	用静态栈数据结构实现表达式求值	214
9.7	哈夫曼编译器	218
	参考文献	226

数据作为计算机加工处理的对象,如何在计算机中表示、存储是计算机科学研究的主要内容之一,更是计算机技术需要解决的关键问题之一。数据是计算机化的信息,是计算机处理的主要对象。科学计算、数据处理、过程控制、文件存储等,都是对数据进行加工处理的过程。因此,要设计出一个结构好、效率高的程序,必须研究数据的特性、数据间的相互关系及其对应的存储表示,并利用这些特性和关系设计相应的算法和程序。

1.1 从问题到程序

“数据结构”作为计算机科学与技术专业的专业基础课,是十分重要的核心课程,其主要研究内容是数据之间的逻辑关系和物理实现,即探索有利的数据组织形式及存取方式。计算机系统软件和应用软件的设计、开发要用到各种类型的数据结构,因此,要想更好地运用计算机来解决实际问题,仅仅依赖几种计算机程序设计语言是不够的,还必须学习和掌握好数据结构的有关知识。

在计算机发展的初期,人们使用计算机的目的主要是处理数值计算问题。当我们使用计算机来解决一个具体问题时,一般需要经过下列几个步骤:首先要从该具体问题抽象出一个适当的数学模型,然后设计或选择一个解此数学模型的算法,再编写程序并进行调试、测试,最后运行程序并得到答案(如图 1.1 所示)。例如,求解梁架结构中应力数学模型的线性方程组,该方程组可以使用迭代算法来求解。

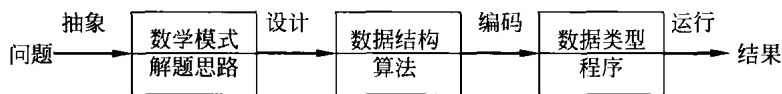


图 1.1 用计算机解决问题的一般过程

由于当时所涉及的运算对象是简单的整型、实型或布尔类型数据,所以程序设计者的主要精力集中于程序设计的技巧上,而不太重视数据结构。随着计算机应用领域的扩大和硬件的发展,非数值计算问题显得越来越重要。据统计,当今处理非数值计算性问题占用了 90% 以上的机器时间。这类问题涉及的处理对象不再是简单的数据类型,其形式更加多样,结构更为复杂,数据元素之间的相互关系一般无法直接用数学方程式加以描述。因此,解决这类问题的关键不再是数学分析和计算方法,而是要设计出合适的数据结构,以便有效地解决问题。

例 1.1 图书信息检索系统。

在现代图书馆中,往往借助计算机图书检索系统来查找需要的图书信息;或者直接通

过图书馆信息系统进行图书借阅。为此,需要将图书信息按不同分类进行编排,建立合适的数据结构进行存储和管理,按照某种算法编写相关程序,实现计算机自动检索。由此,一个简单的图书信息检索系统包括一张按图书分类号和登录号顺序排列的图书信息表,以及分别按作者、出版社等顺序排列的各类索引表,如图 1.2 所示。由这些表构成的文件便是图书信息检索的数学模型。计算机的主要操作便是按照用户的要求(如给定作者)通过不同的索引表对图书信息进行检索、查询。

序号	图书分类号	登录号	书名	作者	出版社
1	B259.1	3240	梁启超家书	张品兴	中国文联出版社
2	C52	5231	探寻语碎	李泽厚	上海文艺出版社
3	D035.5	6712	市政学	张永桃	高等教育出版社
4	G206	1422	传播学	邵培仁	高等教育出版社
5	H319.4	1008	英语阅读策略	李宗宏	兰州大学出版社
6	K825.4.00	5819	围棋人生	聂卫平	中国文联出版社
7	P1.00	8810	通向太空之路	邹惠成	科学出版社
8	TN915	7911	通信与网络技术概论	刘云	中国铁道出版社
9	TP312	7623	计算机软件技术基础	王宇川	科学出版社
10	TP393.07	8001	网络管理与应用	张琳	人民邮电出版社
11	Q3.00	2501	普通遗传学	杨业华	高等教育出版社

(a) 图书信息表

作者	序号
邵培仁	4
李泽厚	2
李宗宏	5
刘云	8
聂卫平	6
王宇川	9
杨业华	11
张琳	10
张品兴	1
张永桃	3
邹惠成	7

(b) 作者索引表

出版社	序号
高等教育出版社	3,4,11
科学出版社	7,9
兰州大学出版社	5
人民邮电出版社	10
上海文艺出版社	2
中国铁道出版社	8
中国文联出版社	1,6

(c) 出版社索引表

图 1.2 图书信息检索系统中的数据结构

诸如此类的还有电话自动查号系统、学生信息查询系统、仓库库存管理系统等。在这类数学模型中,计算机处理的对象之间通常存在着的一种简单的线性关系,这类数学模型可称为线性数据结构。

例 1.2 人机对弈问题。

人机对弈是一个古老的人工智能问题,其解题思想是将对弈的策略事先存入计算机。策略包括对弈过程中所有可能的情况以及响应的对策。在决定对策时,根据当前状态,考虑局势发展的趋势作出最有利的选择。因此,计算机操作的对象(数据元素)是对弈过程中的每一步棋盘状态(格局),数据元素之间的关系由比赛规则决定。通常,这个关系不是线性

的,因为从一个格局可以派生出多个格局,所以通常用树型结构来表示。图 1.3 所示的是井字棋的对弈树。

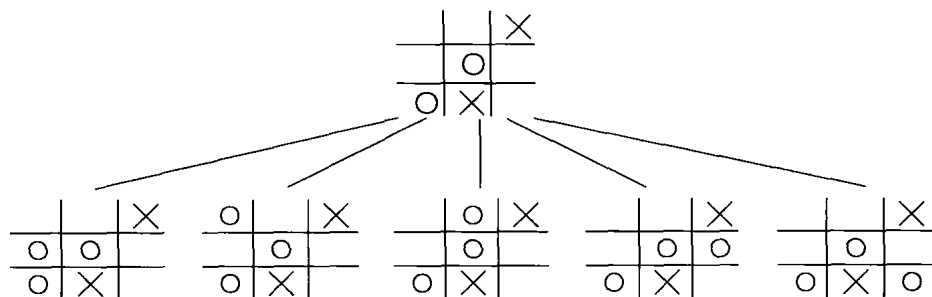


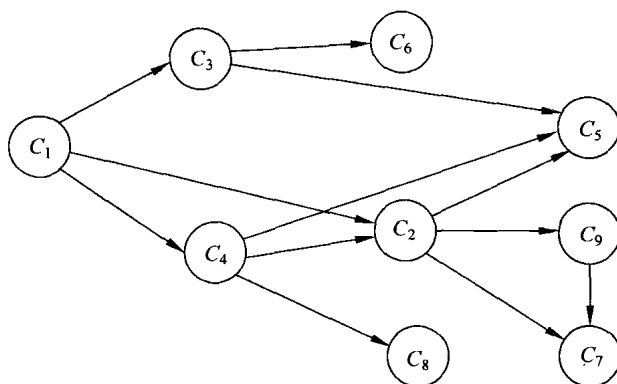
图 1.3 井字棋对弈树

例 1.3 教学计划编排问题。

一个教学计划包含许多课程,在教学计划包含的许多课程之间,有些课程必须按规定的先后次序进行学习,有些则没有次序要求。即有些课程之间有先修和后修的关系。课程之间先修和后修的次序关系可用一个称做“图”的数据结构来表示,如图 1.4 所示。有向图中的每个顶点表示一门课程,如果从顶点 v_i 到 v_j 之间存在有向边 $\langle v_i, v_j \rangle$,则表示课程 i 必须先于课程 j 进行。

课程编号	课程名称	先修课程
C_1	计算机导论	无
C_2	数据结构	C_1, C_4
C_3	汇编语言	C_1
C_4	C 程序设计语言	C_1
C_5	计算机图形学	C_2, C_3, C_4
C_6	接口技术	C_3
C_7	数据库原理	C_2, C_9
C_8	编译原理	C_4
C_9	操作系统	C_2

(a) 计算机专业的课程设置



(b) 表示课程之间优先关系的有向图

图 1.4 教学计划编排问题的数据结构

由以上几个例子可见,描述非数值计算问题的数学模型不再是数学方程,而是诸如表、树、图之类的数据结构。因此,数据结构课程是研究非数值计算的程序设计问题中计算机处理对象以及它们之间关系和操作的学科。

学习数据结构的目的是了解和掌握计算机处理对象的特性,将实际问题中所涉及的处理对象在计算机中表示出来并对它们进行处理。同时,可通过算法训练来提高学生的思维能力,通过程序设计的技能训练来促进学生的综合应用能力和专业素质的提高。

1.2 有关概念和术语

在系统地学习数据结构知识之前,先对一些基本概念和术语赋予确切的定义。

1. 数据

数据(data)是信息的载体,它能够被计算机识别、存储和处理。数据是计算机程序加工的原料。应用程序能处理各种各样的数据,包括数值数据和非数值数据。数值数据是一些整数、实数或复数;非数值数据包括字符、文字、图形、图像、语音等。

2. 数据元素

数据元素(data element)是数据的基本单位,在计算机程序中通常作为一个整体进行考虑和处理。一个数据元素可由若干个数据项组成。在不同的条件下,数据元素又可称为元素、结点、顶点、记录等。例如,学生信息检索系统中学生信息表中的一个记录,教学计划编排问题中的一个顶点等,都被称为一个数据元素。

3. 数据项

数据项(data item)指不可分割的、具有独立意义的最小数据单位,数据项有时也称为字段(Field)或域。例如,学籍管理系统中学生信息表的每一个数据元素就是一个学生记录。它包括学生的学号、姓名、性别、籍贯、出生年月、成绩等数据项。这些数据项可以分为两种:一种叫做初等项,如学生的性别、籍贯等,这些数据项是在数据处理时不能再分割的最小单位;另一种叫做组合项,如学生的成绩,它可以再划分为数学、物理、化学等更小的项。通常,在解决实际问题时把每个学生记录当作一个基本单位进行访问和处理。

4. 数据结构

数据结构(data structure)是指互相之间存在着一种或多种关系的数据元素的集合。在任何问题中,数据元素都不会是孤立的,在它们之间都存在着这样或那样的关系,这种数据元素之间存在的关系称为数据的逻辑结构。根据数据元素之间关系的不同特性,通常有以下4类基本的逻辑结构。

(1) 集合结构:在该结构中,数据元素之间的关系是“属于同一个集合”。数据元素之间除了同属一个集合外,不存在其他关系。

(2) 线性结构:在该结构中,数据元素除了同属于一个集合外,数据元素之间还存在着一对一的顺序关系。

(3) 树型结构:该结构的数据元素之间存在着对多的层次关系。

(4) 图状结构:该结构的数据元素之间存在着多对多的任意关系。图状结构也称做网状结构。

上述4类基本结构的示意图如图1.5所示。

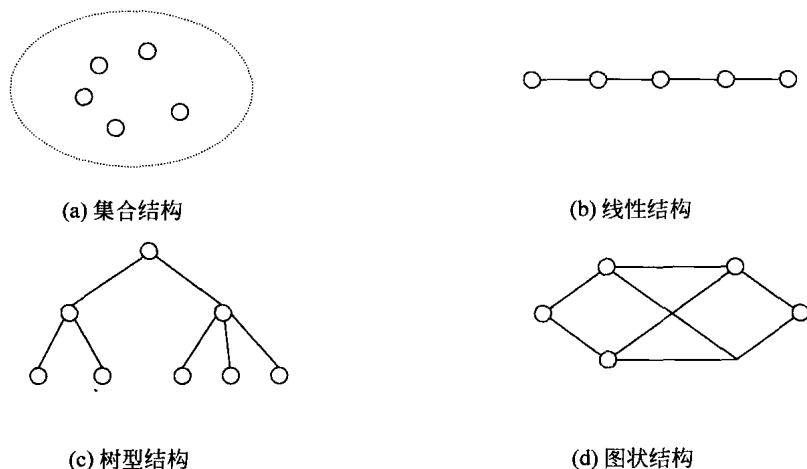


图 1.5 4 类基本结构的示意图

由于集合是数据元素之间极为松散的一种结构,本书不作专门讨论。因此,本书主要讨论线性结构(表、栈、队、串等)和非线性结构(树、图或网)。

从上面所介绍的数据结构的概念中可以知道,一个数据结构有两个要素:一是数据元素,二是数据元素之间的关系。因此,数据结构通常可以采用一个二元组来表示:

$$\text{Data_Structure}=(D,R)$$

其中, D 是数据元素集合, R 是 D 中元素之间关系的集合。

例 1.4 假设一个数据结构定义如下:

$$DS=(D,R)$$

$$D=\{a,b,c,d,e,f,g\}$$

$$R=\{\langle a,b\rangle,\langle a,c\rangle,\langle a,d\rangle,\langle c,e\rangle,\langle c,f\rangle,\langle d,g\rangle\}$$

则该结构的逻辑示意如图 1.6 所示,显然是一个树型结构。

数据结构包括数据的逻辑结构和物理结构。数据的逻辑结构可以看做从具体问题抽象出来的数学模型,它与数据的存储无关。数据的逻辑结构在计算机中的存储表示(又称映像)称为数据的物理结构,或称存储结构。它所研究的是数据结构在计算机中的实现方法,包括数据结构中数据元素的存储表示以及数据元素之间关系的表示。

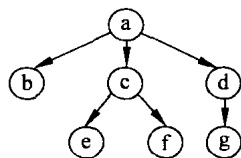


图 1.6 例 1.4 的数据结构逻辑示意图

在计算机中,数据的存储方法包括顺序存储和链式存储。

顺序存储方法通过数据元素在计算机中存储位置关系来表示元素间的逻辑关系,通常把逻辑上相邻的元素存储在物理位置相邻的存储单元中,该存储方式称为顺序存储结构。顺序存储结构是一种最基本的存储表示方法,通常借助于程序设计语言中的数组来实现。

链式存储方法对逻辑上相邻的元素不要求其物理位置相邻,元素间的逻辑关系通过预设的指针字段来表示,由此得到的存储表示称为链式存储结构。链式存储结构通常借助于程序设计语言中的指针类型来实现。

除了顺序存储方法和链式存储方法外,有时为了查找方便还采用索引存储方法和散列

表(Hash)存储方法。

讨论数据结构的目的是为了在计算机中实现对数据的操作,因此在讨论数据的组织结构时必然要考虑在该结构上进行的操作(或称运算)。事实上,数据结构是专门研究某一类数据的表示方法及其相关操作实现算法的一门学科。

5. 数据类型

数据类型(data type)是和数据结构密切相关的一个概念。在高级程序设计语言中用以限制变量取值范围和可能进行的操作的总和称为数据类型。因此所谓数据类型,一是限定了数据的取值范围(实际上与存储形式有关),二是规定了数据能够进行的一组操作(运算)。

数据类型可分为两类:一类是非结构的原子类型,原子类型的值是不可再分解的,如C语言中的基本类型(整型、实型、字符型及指针类型和空类型);另一类是结构类型,它的成分可以由多个结构类型组成,并可以分解。结构类型的成分中可以是非结构的,也可以是结构的。例如数组的值由若干分量组成,每个分量可以是整数等基本类型,也可以是数组等结构类型。

6. 抽象数据类型

抽象数据类型(Abstract Data Type, ADT)是指一个数学模型以及定义在该模型上的一组操作。抽象数据类型的定义取决于它的一组逻辑特性,而与其在计算机内部如何表示和实现无关。即不论其内部结构如何变化,只要它的数学特性不变,都不影响其外部的使用。

抽象数据类型和数据类型实质上是一个概念。例如,各种计算机都拥有的整数类型就是一个抽象数据类型,尽管它们在不同处理器上的实现方法可以不同,但由于其定义的数学特性相同,在用户看来都是相同的。因此,“抽象”的意义在于数据类型的数学抽象特性。

抽象数据类型的定义可以由一种数据结构和定义在其上的一组操作组成,而数据结构又包括数据元素及元素间的关系,因此抽象数据类型一般可以由元素、关系及操作三种要素来定义。

本书在讨论各种数据结构时,就是针对其逻辑结构和具体的存储结构给出相应的数据类型,并在确定的数据类型上通过各种算法实现其各种操作。

1.3 算法及算法分析

算法与数据结构的关系非常紧密,在算法设计时总是先要确定相应的数据结构,而在讨论某一种数据结构时也必然会涉及相应的算法。下面就从算法特性、算法描述、算法分析等三个方面对算法进行介绍。

1.3.1 算法特性

算法(algorithm)是对特定问题求解步骤的一种描述,是指令的有限序列。其中每一条指令表示一个或多个操作。一个算法应该具有下列特性。

- (1) **有穷性**: 一个算法必须在有穷步之后结束,即必须在有限时间内完成。
- (2) **确定性**: 算法的每一步必须有确切的定义,无二义性,且在任何条件下算法只有唯

——一条执行路径,即对于相同的输入只能得出相同的输出。

(3) **可行性**: 算法中的每一步都可以通过已经实现的基本运算的有限次执行得以实现。

(4) **输入**: 一个算法具有零个或多个输入,这些输入取自特定的数据对象集合。

(5) **输出**: 一个算法具有一个或多个输出,这些输出同输入之间存在某种特定的关系。

算法的含义与程序十分相似,但又有区别。一个程序不一定满足有穷性。例如,操作系统,只要整个系统不遭破坏,它将永远不会停止,即使没有作业需要处理,它仍处于动态等待中。因此,操作系统不是一个算法。另一方面,程序中的指令必须是机器可执行的,而算法中的指令则无此限制。算法代表了对问题的求解方法,而程序则是算法在计算机上的特定的实现。一个算法若用程序设计语言来描述,则它就是一个程序。

算法与数据结构是相辅相成的。解决某一类特定问题的算法可以选择不同的数据结构,而且选择恰当与否直接影响算法的效率;反之,一种数据结构的优劣由各种算法的执行效果来体现。

在设计算法时通常需要考虑以下几个方面的要求。

(1) **正确性**: 算法的执行结果应当满足预先规定的功能和性能要求。正确性要求表明算法必须满足实际需求,达到解决实际问题的目标。

(2) **可读性**: 一个算法应当思路清晰、层次分明、简单明了、易读易懂。可读性要求表明算法主要是人与人之间交流解题思路 and 进行软件设计的工具,因此可读性必须要强。同时一个可读性强的算法,其程序的可维护性、可扩展性都要好许多,因此,许多时候人们往往在一定程度上以牺牲效率来提高可读性。

(3) **健壮性**: 当输入不合法数据时,应能作适当处理,不至于引起严重后果。健壮性要求表明算法要全面细致地考虑所有可能的边界情况,并对这些边界条件作出完备的处理,尽可能使算法没有意外的情况。

(4) **高效性**: 有效使用存储空间和有较好的时间效率。高效性主要是指时间效率,即解决相同规模的问题时间尽可能短。

一般来说,数据结构上的基本操作主要有以下几种。

(1) **查找**: 寻找满足特定条件的数据元素所在的位置。

(2) **读取**: 读出指定位置上数据元素的内容。

(3) **插入**: 在指定位置上添加新的数据元素。

(4) **删除**: 删去指定位置上对应的数据元素。

(5) **更新**: 修改某个数据元素的值。

1.3.2 算法描述

算法的描述方法很多,根据描述方法的不同,大致可将算法描述分为以下4种。

(1) **自然语言算法描述**: 用人类自然语言(如中文、英文等)来描述算法,同时还可插入一些程序设计语言中的语句来描述,这种方法也称为非形式算法描述。其优点是不需要专门学习,任何人可以直接阅读和理解,但直观性很差,复杂的算法难写难读。

(2) **框图算法描述**: 这是一种图示法,可以采用方框图、流程图、N-S图等来描述算法,这种描述方法在算法研究的早期曾流行过。它的优点是直观、易懂,但用来描述比较复杂的