



21世纪高职高专规划教材·计算机系列



数据结构

(C语言描述)

方贤进 主编

国防科技大学出版社

21 世纪高职高专规划教材

计算机系列

数据结构

(C 语言描述)

方贤进 主编

国防科技大学出版社

【内容简介】本书是为高职高专计算机及相关专业编写的教材。

本书以 C 语言为程序设计语言,采用系列式的叙述方式,引导读者循序渐进地掌握线性表,栈和队列,数组、串与广义表,树,图等不同的数据结构,并系统地介绍了查找和排序的各种实现方法。

本书为高职高专院校计算机及相关专业的教材,也可供广大计算机爱好者学习使用。

图书在版编目(CIP)数据

数据结构:C 语言描述/方贤进主编. —长沙:

国防科技大学出版社,2010.9

ISBN 978-7-81099-803-1

I. ①数… II. ①方… 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. ①TP311.12②TP312

中国版本图书馆 CIP 数据核字(2010)第 166733 号

出版发行:国防科技大学出版社

网 址: <http://www.gfkdcbs.com>

责任编辑:徐 飞 特约编辑:朱秀英

印 刷 者:北京振兴源印务有限公司

开 本:787mm×1 092mm 1/16

印 张:16.25

字 数:406 千字

版 次:2010 年 9 月第 1 版

印 次:2010 年 9 月第 1 次印刷

定 价:25.00 元

21 世纪高职高专规划教材·计算机系列

编审委员会

- 顾问** 郑启华 清华大学教授
计算机教育资深专家
- 主任** 黄维通 清华大学计算机科学与技术系
全国计算机基础教育研究会副秘书长
- 副主任** 李俊 清华大学信息科学技术学院
骆海峰 北京大学软件与微电子学院
梁振方 上海交通大学电子信息与电气工程学院

委员 (以姓氏笔画为序)

卫世浩	王玉芬	王军号	王建平	卢云宏
付俊辉	朱广丽	刘庆杰	刘春霞	江枫
李永波	李光杰	李克东	李学勇	张春飞
张岩	郑义	姚海军	高国红	徐桂保
殷晓波	程华安	谢广彬	詹林	

课程审定 张歆 清华大学信息科学技术学院
战扬 北京大学软件与微电子学院

内容审定 倪铭辰 清华大学信息科学技术学院
谢力军 北京大学软件与微电子学院
李振华 北京航空航天大学计算机学院

出版说明

高职高专教育作为我国高等教育的重要组成部分,承担着培养高素质技术、技能型人才的重任。近年来,在国家和社会的支持下,我国的高职高专教育取得了不小的成就,但随着我国经济的腾飞,高技能人才的缺乏越来越成为影响我国经济进一步快速健康发展的瓶颈。这一现状对于我国高职高专教育的改革和发展而言,既是挑战,更是机遇。

要加快高职高专教育改革的步伐,就必须对课程体系和教学模式等问题进行探索。在这个过程中,教材的建设与改革无疑起着至关重要的基础性作用,高质量的教材是培养高素质人才的保证。高职高专教材作为体现高职高专教育特色的知识载体和教学的基本工具,直接关系到高职高专教育能否为社会培养并输送符合要求的高技能人才。

为促进高职高专教育的发展,加强教材建设,教育部在《关于全面提高高等职业教育教学质量的若干意见》中,提出了“重点建设好3000种左右国家规划教材”的建议和要求,并对高职高专教材的修订提出了一定的标准。为了顺应当前我国高职高专教育的发展潮流,推动高职高专教材的建设,我们精心组织了一批具有丰富教学和科研经验的人员成立了21世纪高职高专规划教材编审委员会。

编审委员会依据教育部高教司制定的《高职高专教育基础课程教学基本要求》和《高职高专教育专业人才培养目标及规格》,调研了百余所具有代表性的高等职业技术学院和高等专科学校,广泛而深入地了解了高职高专的专业和课程设置,系统地研究了课程的体系结构,同时充分汲取各院校在探索培养应用型人才方面取得的成功经验,并在教材出版的各个环节设置专业的审定人员进行严格审查,从而确保了整套教材“突出行业需求,突出职业的核心能力”的特色。

本套教材的编写遵循以下原则:

(1)成立教材编审委员会,由编审委员会进行教材的规划与评审。

(2)按照人才培养方案以及教学大纲的需要,严格遵循高职高专院校各学科的专业规范,同时最大程度地体现高职高专教育的特点及时代发展的要求。因此,本套教材非常注重培养学生的实践技能,力避传统教材“全而深”的教学模式,将“教、学、做”有机地融为一体,在教给学生知识的同时,强化了对学生实际操作能力的培养。

(3)教材的定位更加强调“以就业为导向”,因此也更为科学。教育部对我国的高职高专教育提出了“以应用为目的,以必需、够用为度”的原则。根据这一原则,本套教材在编写过程中,力求从实际应用的需要出发,尽量减少枯燥、实用性不强的理论灌输,充分体现“以行业为导向,以能力为本,以学生为中心”的风格,从而使本套教材更具实用性和前瞻性,与就业市场结合也更为紧密。

(4)采用“以案例导入教学”的编写模式。本套教材力图突破陈旧的教育理念,在讲解的过程中,援引大量鲜明实用的案例进行分析,紧密结合实际,以达到编写实训教材的

目标。这些精心设计的案例不但可以方便教师授课,同时又可以启发学生思考,加快对学生实践能力的培养,改革人才的培养模式。

本套教材涵盖了公共基础课系列、财经管理系列、物流管理系列、电子商务系列、计算机系列、电子信息系列、机械系列、汽车系列和化学化工系列的主要课程。目前已经规划的教材系列名称如下:

财经管理系列

- 财经管理基础课
- 工商管理类
- 财务会计类
- 经济贸易类
- 财政金融类
- 市场营销类

机械系列

- 机械基础课
- 机械设计与制造类
- 数控技术类
- 模具设计与制造类
- 机电一体化类

计算机系列

- 公共基础课
- 计算机专业基础课
- 计算机网络技术类
- 计算机软件技术类
- 计算机应用技术类

公共基础课系列

物流管理系列

电子商务系列

电子信息系列

化学化工系列

汽车系列

对于教材出版及使用过程中遇到的各种问题,欢迎您通过电子邮件及时与我们取得联系(联系方式详见“教师服务登记表”)。同时,我们希望有更多经验丰富的教师加入到我们的行列当中,编写出更多符合高职高专教学需要的高质量教材,为我国的高职高专教育作出积极的贡献。

21 世纪高职高专规划教材编审委员会

序

21 世纪是科技和经济高速发展的重要时期。随着我国经济的持续快速健康发展, 各行各业对高技能专业型人才的需求量迅速增加, 对人才素质的要求也越来越高。高职高专教育作为我国高等教育的重要组成部分, 在加快培养高技能专业型人才方面发挥着重要的作用。

与国外相比, 我国高职高专教育起步时间短, 这种状况与我国经济发展对人才大量需求的现状是很不协调的。因此, 必须加快高职高专教育的发展步伐, 提高应用型人才的培养水平。

高职高专教育水平的提高, 离不开课程体系的完善。相关领域人才的培养需要一批兼具前瞻性和实践性的优秀教材。教育部高教司针对高职高专教育人才培养模式提出了“以就业为导向”的指导思想, 这也正是本套高职高专教材的编写宗旨和依据。

如何使高职高专教材既突出行业的需求特点, 又突出职业的核心能力? 这是教材编写的过程中必须首先解决的问题。本系列教材编委会深入研究了高职高专教育的课程和专业设置, 并对以往的教材进行了详细分析和认真考察, 力图在不破坏教材系统性的前提下, 加强教材的创新和实践性内容, 从而确保学生在学习专业知识的同时多动手, 增强自己的实践能力, 以加强“知”与“行”的结合。

同时, 本系列教材在编写过程中还充分重视群体和类别的差异性, 面对不同学校和专业方向的定位差异, 精心设计了与其相配套的辅助实验指南及相关的习题解答等。这些栏目的设计使本系列教材内容更加丰富, 条理更为清晰, 为老师的讲授和学生的学习都提供了很大的便利。

经过编委会的辛勤努力, 本套教材终于顺利出版了, 相信本套教材一定能够很好地适应现代高职高专教育的教学需求, 也一定能够在高职高专教育计算机课程的改革中发挥积极的推动作用, 为社会培养更多优秀的应用型人才。

全国计算机基础教育研究会副秘书长

黄维通

前 言

数据结构是计算机及相关专业的一门重要的专业基础课,它的主要任务是讨论数据的各种逻辑结构、物理存储结构以及相关的算法,通过对数据结构课程的学习,使读者能够根据实际需要选择合适的数据结构和算法进行程序设计。

本书从数据结构的教學要求出发,以培养读者的基本专业素质及综合应用能力为目标,重点在于使读者通过数据结构的思维,掌握程序设计中各种算法实现的细节和编程技巧,提高运用知识解决实际问题的能力。

全书共分9章。第1章综述数据结构的基本概念,并对算法的描述方法、时间复杂度和空间复杂度通过实例予以讲述。第2章和第3章讨论了线性数据结构中的线性表、栈、队列及其应用。第4章介绍了数组的相关概念和存储结构、矩阵的压缩存储、串和广义表。第5章和第6章介绍了树和图这两种数据结构及其应用。第7章和第8章讨论了查找和排序,介绍了各种实现方法,并从时间上进行定性和定量的分析和比较。第9章给出了数据结构经典算法的C语言实现方法并提供了相应的指导以供读者参考。

本书具有以下特色:

(1)本书采用结合自然语言的类C语言作为数据结构和算法的描述语言,既便于理解和表达,也便于程序的实现,而且从C语言中精选了一个核心子集,并添加C++引用调用参数传递方式等,使得书中的定义和实现既不拘泥于C语言的细节又容易转换成可执行的C或C++程序。

(2)本书重视实践环节,每章都结合该章的内容,设计了习题并单独列出一节讲解实例,第9章列出了8个上机实验,其中包括“八皇后问题”、“二叉树的遍历”、“旅行商问题”、“哈夫曼编码”等经典问题,通过实验和练习,使读者加深对所学知识的理解和运用,积累编程经验。

(3)本书中的有关程序及第9章的实验均在Visual C++ 6.0及Visual Studio 6.0中运行验证,以确保和数据结构上机环境相一致。

本书由方贤进主编。在编写本书时,编者阅读了大量的同类教材,借鉴了其中的观点方法,博采众长,在此对参考文献中提到和未提到的作者表示感谢。

本书的作者虽从事多年教学工作,但是由于水平和经验所限,时间仓促,书中难免会有疏漏之处,欢迎读者批评指正。

编 者

目 录

第 1 章 绪论	1
1.1 引言	1
1.2 基本术语	3
1.2.1 数据及相关概念	3
1.2.2 数据的逻辑结构	4
1.2.3 数据的存储结构	5
1.2.4 数据类型	6
1.3 算法分析	9
1.3.1 算法的概念及描述	9
1.3.2 算法的复杂度分析	11
1.3.3 算法的设计要求	14
1.4 实例解析	15
本章小结	17
习题 1	17
第 2 章 线性表	19
2.1 线性表的基本概念	19
2.1.1 线性表的定义及特性	19
2.1.2 线性表的抽象数据类型	20
2.2 线性表的顺序存储及操作	21
2.2.1 顺序表	21
2.2.2 顺序表的基本操作	23
2.2.3 顺序表基本操作的算法分析	25
2.3 线性表的链式存储及操作	26
2.3.1 单链表及其基本操作	26
2.3.2 循环链表及其基本操作	34
2.3.3 双向链表及其基本操作	35
2.4 各种存储结构的线性表的比较	38
2.5 实例解析与编程实现	39
本章小结	42
习题 2	43
第 3 章 栈和队列	44
3.1 栈	44
3.1.1 栈的基本概念	44

3.1.2	栈的顺序存储及顺序栈的操作	45
3.1.3	栈的链式存储及链栈的操作和应用	49
3.1.4	栈的简单应用与递归	52
3.2	队列	54
3.2.1	队列的基本概念	54
3.2.2	队列的顺序存储、操作及应用	55
3.2.3	队列的链式存储及链队列的操作	60
3.3	实例解析与编程实现	63
	本章小结	67
	习题3	68

第4章 数组、串与广义表

4.1	数组的基本概念与特性	69
4.2	数组的顺序存储结构及其操作	70
4.3	矩阵的压缩存储	72
4.3.1	特殊矩阵的压缩存储	73
4.3.2	稀疏矩阵的压缩存储	75
4.4	串	81
4.4.1	串的基本概念	81
4.4.2	串的基本操作简述	82
4.4.3	串的存储及基本操作实现	84
4.4.4	串的模式匹配	89
4.5	广义表	94
4.5.1	广义表的定义	94
4.5.2	广义表的存储结构	95
4.5.3	广义表的基本运算	97
4.6	实例解析与编程实现	98
	本章小结	101
	习题4	102

第5章 树

5.1	树的基本概念	103
5.1.1	树的定义	103
5.1.2	有关树的基本术语	104
5.2	二叉树	105
5.2.1	二叉树的基本概念	105
5.2.2	二叉树的性质	107
5.2.3	二叉树的存储结构	109
5.2.4	二叉树的遍历和线索二叉树	111
5.3	树和森林	122

5.3.1	树的存储结构	122
5.3.2	树、森林与二叉树的关系	126
5.4	哈夫曼树	129
5.4.1	哈夫曼树的基本概念	129
5.4.2	哈夫曼树的构造算法	130
5.4.3	哈夫曼编码及算法实现	132
5.5	实例解析与编程实现	133
	本章小结	136
	习题 5	137
第 6 章 图		138
6.1	图的定义和术语	138
6.2	图的存储结构	141
6.2.1	邻接矩阵	141
6.2.2	邻接表	143
6.3	图的遍历	145
6.3.1	深度优先搜索	146
6.3.2	广度优先搜索	147
6.3.3	图的连通分量计算	149
6.4	图的应用	150
6.4.1	最小生成树	150
6.4.2	最短路径	153
6.4.3	拓扑排序	155
6.5	实例解析与编程实现	157
	本章小结	164
	习题 6	165
第 7 章 查找表		166
7.1	基本概念与术语	166
7.2	顺序表的查找	167
7.2.1	顺序查找	167
7.2.2	有序表的折半查找	168
7.2.3	斐波那契查找	171
7.2.4	分块查找	171
7.3	动态查找表	173
7.3.1	二叉排序树	173
7.3.2	二叉排序树中插入结点和构造二叉排序树	174
7.3.3	二叉排序树中删除结点	176
7.4	哈希表	179
7.4.1	哈希表与哈希方法	179

7.4.2 常用的构造哈希函数的方法	180
7.4.3 处理冲突的方法	181
7.4.4 哈希表的查找分析	183
7.5 实例解析	184
本章小结	187
习题 7	187
第 8 章 排序	188
8.1 基本概念	188
8.2 插入排序	190
8.2.1 直接插入排序	190
8.2.2 希尔排序	192
8.3 交换排序	193
8.3.1 冒泡排序	194
8.3.2 快速排序	196
8.4 选择排序	198
8.4.1 简单选择排序	199
8.4.2 堆排序	200
8.5 归并排序	203
8.6 基数排序	205
8.6.1 多关键字排序	205
8.6.2 链式基数排序	206
8.7 外部排序	209
8.7.1 外部排序的方法	209
8.7.2 多路平衡归并的实现	211
8.8 实例解析与编程实现	212
本章小结	217
习题 8	218
第 9 章 实验	219
实验 1 多项式的数组表示	219
实验 2 串的匹配算法及实现	221
实验 3 八皇后问题	224
实验 4 二叉树的遍历	226
实验 5 旅行商问题	231
实验 6 哈夫曼编码	235
实验 7 快速排序	238
实验 8 折半查找	240
参考文献	244

第 1 章 绪 论

数据结构是计算机专业的一门专业基础课程,很多后续课程都要用到本课程所涉及的知识。例如,程序设计、编译技术和操作系统等课程都要使用一些基本的数据结构及其相关的算法;本课程讨论的其他一些数据结构,如广义表、集合以及图等也是很多应用领域经常涉及的。本课程的目的是介绍一些最常用的数据结构,阐明数据结构内在的逻辑关系,讨论它们在计算机中的存储表示,并结合各种数据结构,讨论其各种操作的实现算法。

本章将概括地介绍一些基本概念和术语,包括数据和数据结构、数据的逻辑结构和存储结构、数据类型、算法的概念及描述、算法的复杂度分析等。

1.1 引 言

自从 1946 年第一台计算机问世以来,计算机产业飞速发展,计算机的功能不断增加,速度也不断提高。计算机的应用已从最初的科学计算发展到人类生活的各个领域。计算机加工处理的对象也由纯粹的数值数据发展到字符、表格和图像等各种具有一定结构的数据,这就给程序设计带来一些新的问题。为了设计出一个“好”的程序,必须分析待处理的对象的特性以及各对象之间存在的关系,这就是数据结构这门学科形成和发展的背景。

用计算机解决一个具体问题时,一般需要经过以下几个步骤:首先分析实际问题并从中抽象出一个适当的数学模型,然后设计一个解决此数学模型的算法,最后编制出程序上机调试,直至得到最终的解答,其过程如图 1-1 所示。

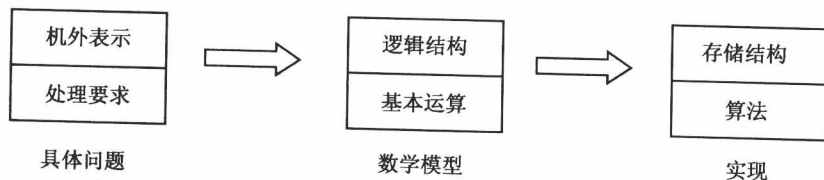


图 1-1 计算机解决具体问题的步骤

寻求数学模型的实质是分析问题,从中提取操作的对象,并找出这些操作对象之间的关系,然后使用数学模型加以描述。数值计算问题的数学模型一般可用数学方程或数学公式来描述,而更多的非数值计算问题无法用数学方程加以描述。下面给出几个例子。

【例 1-1】 某单位职工档案管理问题。为了简单,假定每个职工的档案只包括以下 5 项:职工号、姓名、性别、出生日期和职称。一般来说,档案管理人员很可能将这些档案组织成表格形式,如表 1-1 所示,表中的每一行反映了一个职工的基本情况。

表 1-1 职工档案表

职工号	姓名	性别	出生日期	职 称
0001	曾国庆	男	1965.03.20	高工
0002	吕 晴	女	1966.04.16	助工
0003	许建莉	女	1967.12.09	工程师
0004	张国玉	男	1969.07.28	工程师
0005	关小西	男	1970.09.14	工程师
...	...	...	...	...

本问题中的处理要求包括所有能用计算机完成的档案管理工作,至少应包含以下功能:

- (1)查找,需要在表格中找出某人的档案。
- (2)读取,阅读通过查找找到的档案。
- (3)插入,调入新职工时将该职工的档案加入到表格中。
- (4)删除,某职工调离时,从表格中撤销其档案。
- (5)更新,某职工情况变化(如晋升职称)时,修改档案的有关内容。

上述每一项功能又可称为一个“运算”。由此,从职工档案管理问题抽象出来的数学模型便包含职工档案表和对这表进行的各种运算。在这类管理问题的数学模型中,计算机处理的对象之间通常存在着一种最简单的线性关系,这类数学模型可称为线性的数据结构。诸如此类的还有电话自动查号系统、排课系统、仓库库存管理系统等各种应用系统。

【例 1-2】 UNIX 操作系统中文件系统的系统结构图。UNIX 操作系统中文件系统的底层是根目录,根目录下包含 bin、lib、user、etc 等一级子目录,而 bin 目录下又包含 math、ds、sw 等二级子目录, user 目录下又包含 yin、xie、tang 等二级子目录, yin 目录下有 Linklist.c、Stack.c 和 Tree.c 等若干个文件,如图 1-2 所示。如果将这些目录和文件都画在一张图上,它们之间所呈现的是一种层次关系,从上到下按层进行展开形成一棵倒长的“树”。“树根”是系统的根目录,其他目录和文件是这棵树上的“树叶”,从根结点到某个叶子结点经过的路径就是当前文件的绝对路径。由此可见,“树”也可以是某些非数值计算问题的数学模型,它也是一种数据结构。“树”状结构的模型在生活中接触的也比较多,如国家行政区域规划、书籍目录等。

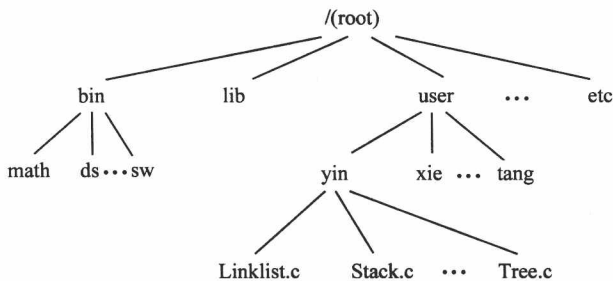


图 1-2 UNIX 操作系统中的文件系统的系统结构图

【例 1-3】 比赛编排问题。有 6 支球队进行足球比赛,分别用 V_1 、 V_2 、 V_3 、 V_4 、 V_5 、 V_6 表示这 6 支球队,它们之间的比赛情况也可以用一个称为“图”的数据结构来表示,图中的每个顶点表示一个球队,如果从顶点 V_i 到 V_j 之间存在有向边 $\langle V_i, V_j \rangle$,则表示球队 V_i 战胜球

队 V_j , 如 V_1 队战胜 V_2 队, V_2 队战胜 V_3 队, V_3 队战胜 V_5 队等, 这种胜负情况可以用图 1-3 表示。

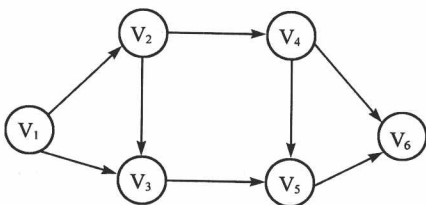


图 1-3 比赛胜负情况图

由此可以看出, 用点、点与点之间的线所构成的图也可以反映实际生产和生活中的某些特定对象之间的特定关系。诸如此类有铁路交通图、教学编排图等。

综合以上 3 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、图之类的数据结构及其运算。因此可以说, 数据结构课程主要是研究非数值性程序设计中所出现的计算机操作对象以及它们之间的关系和运算等的学科。

在美国, 数据结构作为一门独立的课程开始于 1968 年。在这之前, 它的某些内容曾在其他课程中有所阐述。1968 年, 美国一些大学计算机系的教学计划中, 虽然把数据结构规定为一门课程, 但对其内容并未作明确规定。当时, 数据结构几乎和图论, 特别是表、树的理论为同义语。随后, 数据结构的概念被扩充到网络、集合代数论、格、关系等方面, 从而变成了现在称之为“离散结构”的内容。由于数据必须在计算机中进行处理, 所以不仅要考虑数据本身的数学特性, 而且要考虑数据的存储结构, 这就进一步扩充了数据结构的内容。

数据结构在计算机学科中起着承上启下的作用, 在计算机技术的各个领域也有着广泛的应用。学习这门课需要程序设计语言作基础, 学习前要有比较扎实的程序设计基本功, 同时该课程又为后续的数据库等一系列课程奠定基础。数据结构的研究不仅涉及计算机硬件 (特别是编码理论、存储装置和存取方法等) 的研究范围, 而且与计算机软件的研究有着密切的关系, 无论是编译程序还是操作系统, 都涉及到如何组织数据, 使检索和存取数据更为方便。因此可以认为, 数据结构是介于数学、计算机硬件和计算机软件三者之间的一门核心课程。

学习数据结构的目的是了解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。与此同时, 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 基本术语

为了便于以后各章的学习, 本节将对全书中常用到的一些基本概念和术语给以确切的定义。

1.2.1 数据及相关概念

数据(data): 数据是客观事物的符号表示, 在计算机科学中是指所有能输入到计算机中

并被计算机程序处理的符号的总称。数据是计算机进行程序处理的必要的“原料”信息。例如,一个代数方程求解程序所用到的数据是整数和实数,一个编译程序处理的数据是字符串(源程序)。在计算机科学中,数据的含义相当广泛,如表格处理软件中的各类表格信息,多媒体处理软件中的各种多媒体信息,包括音频、视频信息等。总之,数据是通过编码形成的各种客观事物的信息。

数据元素(data element):数据元素是数据的基本单位,在计算机程序中通常把它作为一个整体来处理。例如,【例 1-2】中的文件系统的系统结构图的一个目录,【例 1-3】中的“图”的一个圆圈都被称为一个数据元素。有时,一个数据元素又可以由若干个**数据项**(data item)组成。如表 1-2 所示学生成绩表中的一条记录为一个数据元素,而记录中的学号、姓名、语文等都分别为一个数据项。数据项是数据的不可分割的最小单位。数据元素也可以仅有一个数据项。

表 1-2 学生成绩表

学 号	姓 名	语 文	数 学	C 语言
6201001	张三	85	54	92
6201002	李四	92	84	64
6201003	王五	87	74	73
...	...	...	...	...

数据对象(data object):数据对象是性质相同的数据元素组成的集合,是数据的一个子集。数据元素是数据对象的数据成员。例如,正整数的数据对象是集合 $N=\{1,2,3,4,\dots\}$, 字母字符数据对象是集合 $N=\{'A','B','C',\dots,'Z'\}$ 。

表 1-2 所示的学生成绩表也是一个数据对象。每个学生的记录为一个数据元素。每个数据元素由学号、姓名、语文、数学、C 语言这些数据项组成。

数据结构(data structure):是相互之间存在一种或多种特定关系的数据元素的集合。在任何实际问题中,各数据元素之间都不可能是孤立的,它们之间总是存在着这样或那样的关系,这种数据元素之间的关系就称为结构。数据结构包括数据的逻辑结构和存储结构。

1.2.2 数据的逻辑结构

根据数据元素之间关系的不同,可以把逻辑结构分成以下 4 种形式:

(1)集合:数据元素间为松散的关系。在集合结构中各元素之间,除了“同属于某一个数据对象”的关系外,再无其他的关系,如图 1-4(a)所示。

(2)线性结构:数据元素间为严格的一对一关系,除第一个元素外,其他元素只有一个前驱,除最后一个元素外,其他元素只有一个后继,如图 1-4(b)所示。表 1-2 的学生成绩表也是一个线性结构,在这个结构中,数据元素(由一个人的学号、姓名和其成绩组成)的排列十分有序,第一个元素之后紧跟着第二个元素,第二个元素之后紧跟着第三个元素,以此类推,故称“线性结构”。

(3)树状结构:数据元素之间为严格的一对多的关系,即一个元素只有一个前驱,但可以有多个后继,如图 1-4(c)所示。这种结构像自然界中倒立的“树”,呈分支、层次状态。例如,家谱、行政组织结构等都可用树状结构来表示。

(4)图状结构:数据元素之间存在多对多的关系,也就是说元素间的逻辑关系可以是任意的。图状结构如图 1-4(d)所示。

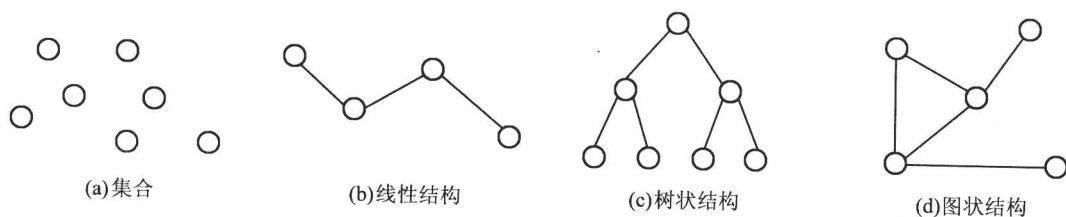


图 1-4 4 种结构示意图

任何数据对象中各数据元素之间都存在特定的关系,而这些存在着相互关系的数据元素的集合就是数据结构。简单地说,数据结构就是带有结构的数据元素的集合。

数据结构可以用二元组来进行数学意义上的形式定义:

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

其中, D 是数据元素的有限集,即数据对象; R 是 D 中所有数据元素之间的关系的有限集。

【例 1-4】为毕业设计小组设计一个数据结构。假设每个小组的关系是 1 名教师指导 1~10 名学生,则数据结构的二元组表示如下:

$$\text{Group} = (P, R)$$

其中, $P = \{T, G_1, \dots, G_n, 1 \leq n \leq 10\}$; $R = \{ \langle T, G_i \rangle \mid 1 \leq i \leq n, 1 \leq n \leq 10 \}$ 。

上述数据结构的定义是一种数学意义上的定义。“数据结构”定义中的“关系”是指数据间的逻辑关系,故也称数据结构为逻辑结构。可将图 1-4 中的集合、树状结构、图状结构归纳为非线性结构。因此,数据的逻辑结构可分为两大类,即线性结构和非线性结构。

然而,讨论数据结构的目的是在计算机中实现对它的操作,因此还需研究如何在计算机中表示它,即数据的存储结构,又称物理结构。

1.2.3 数据的存储结构

数据元素之间的逻辑结构是根据实际问题的需要而选择设计的,因此是面向问题而与计算机本身无关的,或者说不依赖于机器的。而数据元素的存储结构是指数据在计算机中的物理表示和存储的方式,涉及数据元素及其关系在存储器中的物理位置、机器响应的速度等方面的因素,因而物理结构的设计是与计算机本身密切相关的。在计算机中存储信息的最小单位叫做位(bit),8 位可表示一个字节(byte),两个字节称为一个字(word),字节、字或更多的二进制位可称为位串,这个位串称为元素(element)或结点(node)。当数据元素由若干个数据项组成时,则位串中对应于每个数据项的子位串称为数据域(data field)。

对于线性表,其逻辑结构是:除第一个元素外,其他元素只有一个前驱,除最后一个元素外,其他元素只有一个后继。它在计算机中的表示和存储有以下两种方式:用连续的存储单元存储;用分散的存储单元存储,并用指针将其连接。

数据元素之间的关系在计算机中有两种不同的存储方法:顺序存储和链式存储。

顺序存储:其方法是把数据元素依次存储在一组地址连续的存储单元中,元素间的逻辑关系由存储单元的位置直接体现,由此得到的存储表示称为顺序存储。高级语言中,常用一维数组来实现顺序存储。该方法主要用于线性结构,非线性结构也可通过某种线性化的处