

普通高校本科计算机专业特色教材精选·算法与程序设计

数据结构

管致锦 徐 慧 陈德裕 编著

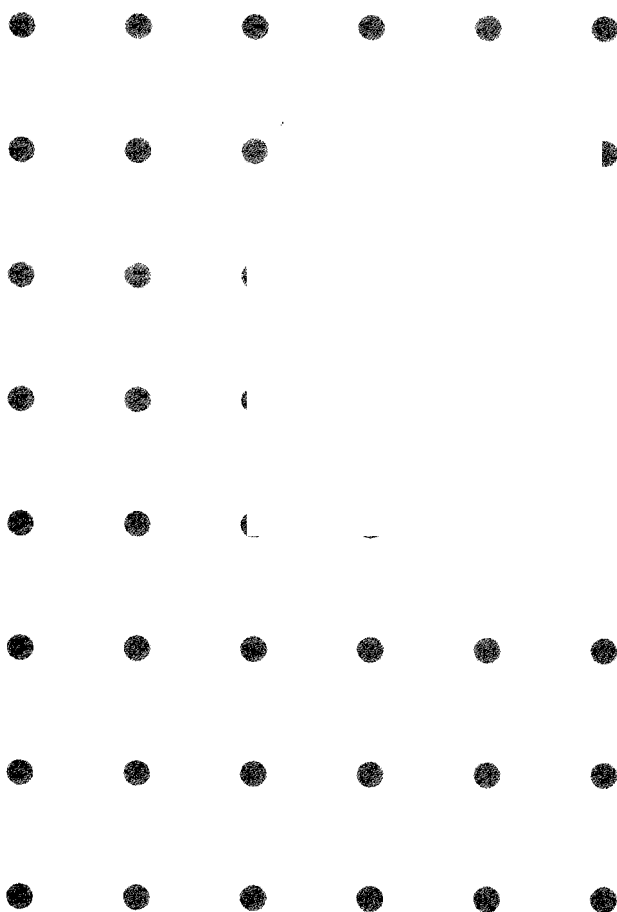
清华大学出版社



普通高校本科计算机专业特色教材精选·算法与程序设计

数据结构

管致锦 徐 慧 陈德裕 编著



清华大学出版社
北京

内 容 简 介

本书根据数据结构的特点,充分考虑到教师教学、学生学习与进一步深造,以及相关人员实际工作需要,在处理好数据结构的组织方式和强化算法的实践与应用的同时,使学生通过实现算法的复杂程序训练,编写出结构清晰、正确易读、符合软件工程规范的程序;使教师方便组织教学内容,教学过程结构清晰,内容循序渐进且易于讲解。

本书符合教育部计算机及相关专业研究生考试大纲对数据结构内容的要求。

本书使用C++作为数据结构和算法的描述语言。采用C++语言中的类来表示抽象数据类型(ADT),用C++的类和面向对象结构实现数据结构的算法。所使用的C++代码在Visual C++编译器上全部通过测试。

为了方便本书的学习和教学,提供有配套教材《数据结构实践教程》、《数据结构学习指导与习题集》和相关的学习课件,本系列教材的所有源代码都可以从清华大学出版社网站(<http://www.tup.com>)上免费下载。

本书可作为计算机类及其相关专业的教材,也可供从事计算机工程与应用的科技工作者参考。

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

数据结构 / 管致锦,徐慧,陈德裕编著. —北京:清华大学出版社,2010.3

(普通高校本科计算机专业特色教材精选·算法与程序设计)

ISBN 978-7-302-21475-5

I. ①数… II. ①管… ②徐… ③陈… III. ①数据结构—高等学校—教材 IV. ①TP311.12

中国版本图书馆CIP数据核字(2010)第026130号

责任编辑:袁勤勇 李玮琪

责任校对:李建庄

责任印制:李红英

出版发行:清华大学出版社

<http://www.tup.com.cn>

社总机:010-62770175

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

地址:北京清华大学学研大厦A座

邮编:100084

邮购:010-62786544

印装者:北京国马印刷厂

经销:全国新华书店

开本:185×260

印张:19

字数:471千字

版次:2010年3月第1版

印次:2010年3月第1次印刷

印数:1~3000

定价:26.00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:035503-01

出版说明

INTRODUCTION

在我国高等教育逐步实现大众化后，越来越多的高等学校将会面向国民经济发展的第一线，为行业、企业培养各级各类高级应用型专门人才。为此，教育部已经启动了“高等学校教学质量和教学改革工程”，强调要以信息技术为手段，深化教学改革和人才培养模式改革。如何根据社会的实际需要，根据各行各业的具体人才需求，培养具有特色显著的人才，是我们共同面临的重大问题。具体地说，培养具有一定专业特色的和特定能力强的计算机专业应用型人才是计算机教育要解决的问题。

为了适应 21 世纪人才培养的需要，培养具有特色的计算机人才，急需一批适合各种人才培养特点的计算机专业教材。目前，一些高校在计算机专业教学和教材改革方面已经做了大量工作，许多教师在计算机专业教学和科研方面已经积累了许多宝贵经验。将他们的教研成果转化为教材的形式，向全国其他学校推广，对于深化我国高等学校的教学改革是一件十分有意义的事情。

清华大学出版社在经过大量调查研究的基础上，决定组织出版一套“普通高校本科计算机专业特色教材精选”。本套教材是针对当前高等教育改革的新形势，以社会对人才的需求为导向，主要以培养应用型计算机人才为目标，立足课程改革和教材创新，广泛吸纳全国各地的高等院校计算机优秀教师参与编写，从中精选出版确实反映计算机专业教学方向的特色教材，供普通高等院校计算机专业学生使用。

本套教材具有以下特点：

1. 编写目的明确

本套教材是在深入研究各地各学校办学特色的基础上，面向普通高校的计算机专业学生编写的。学生通过本套教材，主要学习计算机科学与技术专业的基本理论和基本知识，接受利用计算机解决实际问题的基本训练，培养研究和开发计算机系统，特别是应用系统的基本能力。

2. 理论知识与实践训练相结合

根据计算学科的三个学科形态及其关系,本套教材力求突出学科的理论与实践紧密结合的特征,结合实例讲解理论,使理论来源于实践,又进一步指导实践。学生通过实践深化对理论的理解,更重要的是使学生学会理论方法的实际运用。在编写教材时突出实用性,并做到通俗易懂,易教易学,使学生不仅知其然,知其所以然,还要会其如何然。

3. 注意培养学生的动手能力

每种教材都增加了能力训练部分的内容,学生通过学习和练习,能比较熟练地应用计算机知识解决实际问题。既注重培养学生分析问题的能力,也注重培养学生解决问题的能力,以适应新经济时代对人才的需要,满足就业要求。

4. 注重教材的立体化配套

大多数教材都将陆续配套教师用课件、习题及其解答提示,学生上机实验指导等辅助教学资源,有些教材还提供能用于网上下载的文件,以方便教学。

由于各地区各学校的培养目标、教学要求和办学特色均有所不同,所以对特色教学的理解也不尽一致,我们恳切希望大家在使用教材的过程中,及时地给我们提出批评和改进意见,以便我们做好教材的修订改版工作,使其日趋完善。

我们相信经过大家的共同努力,这套教材一定能成为特色鲜明、质量上乘的优秀教材。同时,我们也希望通过本套教材的编写出版,为“高等学校教学质量和教学改革工程”做出贡献。

清华大学出版社

前言

PREFACE

数据结构是计算机科学各专业以及其他相近专业的核心课程之一，其研究对象为问题求解方法、程序设计方法和典型的数据结构算法。

本书根据现代教育教学特点，充分考虑到一般本科院校学校计算机及其相关专业的现状，严格遵循教育部计算机及相关专业研究生考试大纲的要求，吸收国外教材的一些新思想，既有创新，又兼顾了传统教材的优点。把教师的教学要求与学生学习、实际工作需要和进一步深造等需求紧密结合起来。

本书既考虑数据结构的组织方式，又强化算法的实践与应用。采用自上而下的设计方法，从抽象的数据结构描述到数据结构的组织，再到操作的具体实现，并通过实例说明应用方法。这种组织方法的目的是：为了使学生在过程中更好地分析研究计算机加工数据的特性，以便为应用所涉及的数据选择适当的逻辑结构、存储结构及其相应的操作方法，通过程序设计实现相应的算法，求得相关问题的解，并能掌握算法的时间分析和空间分析技术。使学生通过实现算法的复杂程序训练，编写出结构清晰、正确易读、符合软件工程规范的程序。使教师方便组织教学内容，教学过程结构清晰，内容循序渐进易于讲解。

本书使用标准C++作为数据结构和算法的描述语言。采用C++语言中的类来表示抽象数据类型（ADT）；尽可能用C++的类和面向对象结构来实现数据结构的算法。对基本的数据结构采用面向对象的方法，将数据与对象封装成类，以便在其他应用程序中直接使用。

本书读者应已经修完程序设计课程并熟悉标准C++语言，所使用的C++代码均在VC++编译器上全部通过测试。

本书是多位老师多年来从事数据结构教学的智慧和经验的结晶，提供了丰富的教学辅助手段，其配套教材有《数据结构实践教程》和《数据结构学习指导与习题集》，同时由清华大学出版社出版。这些辅助教材为数据结构课程的教学和学习提供了有效的保障。

本书第1章、第6章和第7章由管致锦编写,第2章~第5章由徐慧编写,第8章和第9章由陈德裕编写,全书的统稿由管致锦负责。杭月芹、顾颀、周建美、丁卫平、陈苏蓉、顾卫江、丁红、章雅娟、周洁、朱颖等对本书进行了详细审阅,并提出了宝贵意见。聂志浪、赵戈杰、凡刚、陈惊雷、禹杰等同学完成了本书大部分算法代码的测试工作。

本书及其配套教材编写和实践过程历时3年,尽管做了大量的努力,也难免存在不妥和错误之处,恳请读者予以指正,我们也会在适当时间进行修订和补充,同时对教材中引用和参考的其他同行的文献资料在此一并致以感谢。

编 者

2009年12月

目 录

CONTENTS

第1章 绪论	1
1.1 数据结构的概念	1
1.1.1 引言	1
1.1.2 数据结构的发展及其在计算机科学中所处的地位	2
1.1.3 什么是数据结构	3
1.1.4 有关概念和术语	3
1.2 数据类型和抽象数据类型	6
1.2.1 数据类型	6
1.2.2 抽象数据类型	6
1.3 算法和算法分析	6
1.3.1 算法特性	7
1.3.2 算法描述	7
1.3.3 算法性能分析与度量	8
第2章 线性表	11
2.1 线性表的类型定义	11
2.2 线性表的顺序存储结构及实现	13
2.2.1 线性表的顺序存储	13
2.2.2 顺序表的实现	14
2.3 线性表的链式存储结构及实现	18
2.3.1 线性表的链式存储	18
2.3.2 单链表的实现	18
2.3.3 其他形式的链表	26
2.4 线性表的其他存储方法	27
2.4.1 顺序存储与链式存储的比较	27
2.4.2 静态链表	28
2.4.3 间接寻址	29
2.5 线性表应用举例	29

第3章 特殊线性表	35
3.1 栈	35
3.1.1 栈的逻辑结构	35
3.1.2 栈的顺序存储结构及实现	37
3.1.3 栈的链式存储及实现	40
3.1.4 顺序栈和链栈的比较	42
3.1.5 栈的应用举例	42
3.2 队列	48
3.2.1 队列的逻辑结构	49
3.2.2 队列的顺序存储结构及实现	50
3.2.3 队列的链式存储及实现	54
3.2.4 队列的应用	57
第4章 串及其模式匹配	59
4.1 串的定义	59
4.1.1 串的相关概念	59
4.1.2 串的抽象数据类型定义	60
4.2 串的存储结构	61
4.2.1 串的顺序存储结构	62
4.2.2 串的链式存储结构	62
4.2.3 串的索引存储结构	63
4.2.4 串的堆存储	64
4.3 顺序串的实现	64
4.3.1 常用C++ 字符串函数	64
4.3.2 串类	65
4.4 串操作举例	71
4.5 模式匹配	72
第5章 广义线性表	79
5.1 数组	79
5.1.1 数组的定义	79
5.1.2 数组的顺序存储	81
5.2 矩阵的压缩存储	83
5.2.1 特殊矩阵的压缩存储	83
5.2.2 稀疏矩阵的压缩存储	85
5.2.3 稀疏矩阵的运算	87
5.3 广义表	97

5.3.1	广义表的逻辑结构	97
5.3.2	广义表的存储	100
5.3.3	广义表的实现	102
第6章	树和二叉树	107
6.1	树的定义和基本术语	107
6.2	二叉树	109
6.2.1	二叉树的基本概念	109
6.2.2	二叉树的主要性质	114
6.3	二叉树的存储结构与实现	116
6.3.1	二叉树的存储	116
6.3.2	二叉树的基本操作及实现	118
6.4	二叉树的遍历	126
6.4.1	二叉树的遍历方法及递归实现	126
6.4.2	二叉树遍历的非递归实现	131
6.4.3	由遍历序列恢复二叉树	134
6.4.4	不用栈的二叉树遍历的非递归方法	136
6.5	线索二叉树	137
6.5.1	线索二叉树及其结构	137
6.5.2	线索二叉树的基本操作实现	138
6.6	二叉树的应用	143
6.6.1	二叉树遍历的应用	143
6.6.2	最优二叉树——哈夫曼树	145
6.7	树	151
6.7.1	树的基本操作	151
6.7.2	树的存储结构	153
6.8	树、森林与二叉树的转换	157
6.8.1	树转换为二叉树	157
6.8.2	森林转换为二叉树	158
6.8.3	二叉树转换为树和森林	159
6.9	树和森林的遍历	159
6.9.1	树的遍历	159
6.9.2	森林的遍历	160
6.10	树的应用	160
6.10.1	判定树	161
6.10.2	集合的表示	162
6.10.3	求关系等价类问题	164

第 7 章 图	167
7.1 图的基本概念	167
7.2 图的存储结构及实现	172
7.2.1 邻接矩阵	172
7.2.2 邻接表	174
7.2.3 十字链表	178
7.2.4 邻接多重表	181
7.3 图的遍历	182
7.3.1 深度优先搜索	183
7.3.2 广度优先搜索	185
7.4 图的连通性	186
7.4.1 无向图的连通性	186
7.4.2 有向图的连通性	187
7.4.3 生成树和生成森林	187
7.4.4 关节点和重连通分量	190
7.5 最小生成树	193
7.5.1 最小生成树的基本概念	193
7.5.2 构造最小生成树的 Prim 算法	194
7.5.3 构造最小生成树的 Kruskal 算法	196
7.6 最短路径	199
7.6.1 从一个源点到其他各点的最短路径	199
7.6.2 每一对顶点之间的最短路径	202
7.7 有向无环图及其应用	204
7.7.1 有向无环图的概念	204
7.7.2 AOV 网与拓扑排序	205
7.7.3 AOE 图与关键路径	210
第 8 章 查找	217
8.1 基本概念与术语	217
8.2 静态查找表	219
8.2.1 静态查找表结构	219
8.2.2 顺序查找	220
8.2.3 有序表的折半查找	222
8.2.4 有序表的其他查找方法	225
8.2.5 分块查找	226
8.3 动态查找表	227
8.3.1 二叉排序树	227
8.3.2 平衡二叉树(AVL 树)	234

8.3.3 B-树和 B+树	240
8.4 哈希表查找(杂凑法、散列法)	245
8.4.1 哈希表与哈希方法	245
8.4.2 常用的哈希函数	246
8.4.3 处理冲突的方法	248
8.4.4 哈希表的查找分析	250
第9章 排序	257
9.1 基本概念	257
9.2 插入排序	258
9.2.1 直接插入排序	258
9.2.2 折半插入排序	260
9.2.3 表插入排序	261
9.2.4 希尔排序	264
9.3 交换排序	266
9.3.1 冒泡排序	267
9.3.2 快速排序	268
9.4 选择排序	271
9.4.1 简单选择排序	272
9.4.2 树形选择排序	273
9.4.3 堆排序	276
9.5 归并排序	279
9.6 基数排序	281
9.6.1 多关键字排序	281
9.6.2 链式基数排序	282
9.7 各种内部排序方法的比较	286
参考文献	288

绪 论

计算机科学是对信息进行表示和处理的科学。计算机中表示和处理的信息是以数据的形式体现的。数据的表示和组织直接关系到计算机程序能否处理这些数据以及处理的效率。因此,为了更有效地设计出高效率、可靠性强的程序,需要研究数据的特性、数据间的相互关系以及数据在计算机内部的存储表示,并利用这些特性和关系设计出相应的算法和程序。

1.1 数据结构的概念

1.1.1 引言

计算机程序是对信息(数据)进行加工处理。一般情况下,这些信息之间都存在着一一定的关系。当计算机程序所涉及的运算对象是简单的整型、实型或布尔类型数据时,程序设计者的主要精力是集中于程序设计的技巧上;当计算机处理非数值计算问题时,这类问题涉及的数据之间关系可能非常复杂,甚至很多问题无法用数学方程式加以描述。一般来说,不会同时处理属性不同且相互之间没有关系的各类数据。

例 1.1 设有一个电话号码表,记录了 n 个人的名字和其对应的电话号码,假定按如下形式安排:

$$(\text{name}_1, \text{number}_1), (\text{name}_2, \text{number}_2), \dots, (\text{name}_n, \text{number}_n)$$

要求设计一个程序,当给定任何一个人的名字时,能够打印出此人的电话号码;如果表中没有这个人,则能够报告没有该人的提示信息。

设计程序解决此问题时,主要依赖于计算机如何存储人的名字和电话号码,或者说依赖于名字和电话号码的组织结构如何构造。如果名字和电话号码在计算机中排列次序没有任何规律,则只能用逐一比较的方法(即线性的方法)进行处理。这种方法相当费时,当电话号码表中的人数相当多的时候(比如几百万),显然这种方法查询的效率是很低的。然而,假如把电话号码表中的数据做适当组织,比如按字典顺序排列名字,名字和相

应的号码对应,问题就容易解决了。例如,要查找某人的电话号码,他的名字的第一个字母是 L,只要查找 L 打头的那些名字就可以了,另外 25 个字母就不需要再查找了。

由此可见,数据的组织结构形式直接影响着程序的执行效率。

上述电话号码表中的数据是一种简单的线性关系,这类数学模型可称为线性的数据结构。

例 1.2 家庭成员的关系问题。构成家庭成员名的集合,如{父亲,儿子,女儿,孙子,孙女},这些数据有一个共同特征,即他们都是家庭的成员名。在考虑家庭成员辈分关系的时候,他们构成了一个树型结构,如图 1-1 所示。

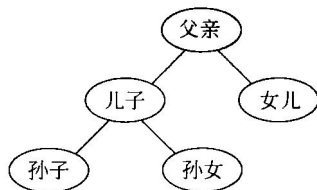


图 1-1 家庭成员关系图

例 1.3 煤气管道的铺设问题。图 1-2 所示为城市的小区之间铺设煤气管道,对 n 个小区只需铺设 $n-1$ 条管线,由于地理环境等不同因素使各条管线所需投资不同(如图上所标识),如何使投资成本最低? 这是一个讨论图的生成树的问题。

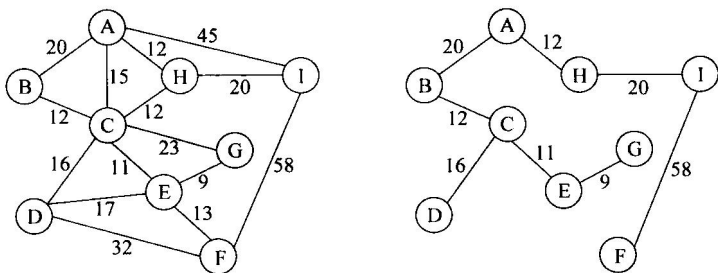


图 1-2 煤气管道的铺设示意图

由于数据的表示方法和组织形式直接关系到程序对数据的处理效率,而系统程序和许多应用程序的规模很大,结构相当复杂,处理对象又多为非数值性数据。因此,单凭程序设计人员的经验和技巧已难以设计出效率高、可靠性强的程序。于是,就要求人们对计算机程序加工的对象进行系统的研究,即研究数据的特性以及数据之间存在的关系——数据结构(data structure)。

1.1.2 数据结构的发展及其在计算机科学中所处的地位

数据结构作为一门独立的课程在国外是从 1968 年才开始的,但在此之前其有关内容已散见于编译原理及操作系统之中。20 世纪 60 年代中期,美国的一些大学开始设立有关课程,但当时的课程名称并不叫数据结构。1968 年美国人 Donald E. Knuth 开创了数据结构的最初体系,他所著的《计算机程序设计的艺术》第一卷《基本算法》是第一本较系统地阐述数据的逻辑结构和存储结构及其操作的著作。从 20 世纪 60 年代末到 70 年代初,出现了大型程序,软件也相对独立,结构程序设计成为程序设计方法学的主要内容,人们越来越重视数据结构。从 20 世纪 70 年代中期到 80 年代,各种版本的数据结构著作相继出现。目前,数据结构的发展并未终结,一方面,面向各专门领域中特殊问题的数据结

构得到研究和发展,如多维图形数据结构等;另一方面,从抽象数据类型和面向对象的观点来讨论数据结构已成为一种新的趋势,越来越被人们所重视。

数据结构与数学、计算机硬件和软件有着十分密切的关系。数据结构是介于数学、计算机硬件和计算机软件之间的一门计算机科学与技术专业的核心课程,是高级程序设计语言、编译原理、操作系统、数据库、人工智能等课程的基础。同时,数据结构技术也广泛应用于信息科学、系统工程、应用数学以及各种工程技术领域。

数据结构课程集中讨论软件开发过程中的设计阶段、设计编码和分析阶段的若干基本问题。此外,为了构造出好的数据结构及其实现,还需考虑数据结构及其实现的评价与选择。因此,数据结构的内容包括3个层次的5个“要素”,如表1-1所示。

表 1-1 数据结构课程内容体系

层次\方面	数据表示	数据处理
抽象	逻辑结构	基本运算
实现	存储结构	算法
评价	不同数据结构的比较及算法分析	

数据结构的核心技术是分解与抽象。通过分解可以划分出数据的3个层次;再通过抽象,舍弃数据元素的具体内容,就得到逻辑结构。类似地,通过分解将处理要求划分成各种功能,再通过抽象舍弃实现细节,就得到运算的定义。上述两个方面的结合使问题变换为数据结构。这是一个从具体(即具体问题)到抽象(即数据结构)的过程。然后,通过增加对实现细节的考虑,进一步得到存储结构和实现运算,从而完成设计任务。这是一个从抽象(即数据结构)到具体(即具体实现)的过程。熟练地掌握这两个过程是数据结构课程在专业技能培养方面的基本目标。

1.1.3 什么是数据结构

计算机解决一个具体问题时,大致需要经过下列几个步骤:首先要从具体问题中抽象出一个适当的数学模型,然后设计一个解此数学模型的算法(algorithm),最后编出程序、进行测试、调整直至得到最终解答。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含有的关系,然后用数学的语言加以描述。

计算机算法与数据的结构密切相关,算法无不依附于具体的数据结构,数据结构直接关系到算法的选择和效率。

运算是由计算机来完成,这就要设计相应的插入、删除和修改的算法。也就是说,数据结构还需要给出每种结构类型所定义的各种运算的算法。

数据结构是研究程序设计中计算机操作的对象以及它们之间的关系和运算的一门学科。

1.1.4 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含义。

数据：数据(data)是信息的载体,是所有能被输入到计算机中,且能被计算机处理的符号(数字、字符等)的集合,它是计算机操作对象的总称。数据是计算机处理信息的某种特定的符号表示形式。用集合的表示方法如下:

$$\text{数据} = \{x | x \text{ 是计算机操作的对象}\}$$

数据包括所有的数字和字符。图形和声音等信息最后也都可以转化为“字符”进行处理。而这些字符和数字是客观信息的一种描述。如 3000 可以表示某个城市的 3000 万人口,也可表示 3000 元/平方米的房价。

数据元素和数据项：数据元素(data element)是数据的基本单位。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,电话号码表中一个人的信息、煤气管道的铺设问题中一个小区的信息等,都被称为一个数据元素。数据元素可分为两类,一类是不可分割的“原子”型数据元素,如:整数“5”,字符“N”等;另一类是由多个款项构成的数据元素,其中每个款项被称为一个“数据项”。例如描述一个学生信息的数据元素可由如图 1-3 所示的 6 个数据项组成。

姓名	学号	性别	班级号	出生日期	入学成绩
----	----	----	-----	------	------

图 1-3 学生信息数据项

其中的出生日期又可以由 3 个数据项:“年”、“月”和“日”组成,则称“出生日期”为“组合项”,而其他不可分割的数据项为“原子项”。数据项是数据结构中讨论的最小单位。

关键字：关键字是指能识别一个或多个数据元素的数据项。若能起唯一识别的作用,则称之为主关键字,否则称之为次关键字。

有时,一个数据元素可由若干个数据项(data item)组成,例如,电话号码表中个人信息中的姓名和号码,煤气管道铺设问题中一个小区的信息中的小区名称、管道长度等。

数据对象：数据对象(data object)或数据元素类(data element class)是具有相同性质的数据元素的集合。在某个具体问题中,数据元素都具有相同的性质(元素值不一定相等,但具有相同的属性),属于同一数据对象(数据元素类),数据元素是数据元素类的一个实例。例如,在煤气管道的铺设问题中,所有的顶点是一个数据元素类,顶点 A 和顶点 B 各自代表一个小区,是该数据元素类中的两个实例,其数据元素的值分别为 A 和 B。

数据结构：数据结构(data structure)是指相互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,数据元素之间都不会是孤立的,在它们之间都存在着这样或那样的关系,这种数据元素之间的关系称为结构。根据数据元素间关系的不同特性,通常有下列 4 类基本的结构:

(1) 集合结构。在集合结构中,数据元素间的关系是“属于同一个集合”。集合是元素关系松散的一种结构。

(2) 线性结构。该结构的数据元素之间存在着一对一的关系。

(3) 树型结构。该结构的数据元素之间存在着一对多的关系。

(4) 图型结构。该结构的数据元素之间存在着多对多的关系,图型结构也称作网状结构。图 1-4 为表示上述 4 类基本结构的示意图。

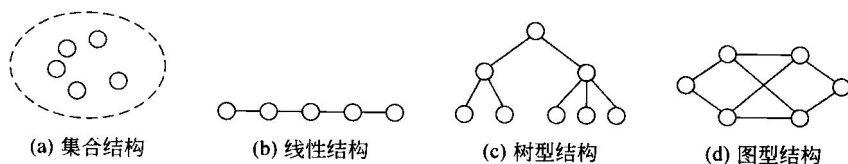


图 1-4 四类基本结构的示意图

由于集合是数据元素之间关系极为松散的一种结构,因此在实际问题解决中往往使用其他结构来表示它。

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。一个是数据元素的集合,另一个是关系的集合。在形式上,数据结构通常可以采用一个二元组来表示。

数据结构的定义形式为:

$$\text{Data_Structure} = (D, R) \quad (1-1)$$

其中, D 是数据元素的有限集, R 是 D 上关系的有限集。

例 1.2 中家庭成员数据结构可以表示成:

$$F = (D, R)$$

$$D = \{\text{父亲, 儿子, 女儿, 孙子, 孙女}\}$$

$$R = \{(\text{父亲, 儿子}), (\text{父亲, 女儿}), (\text{儿子, 孙子}), (\text{儿子, 孙女}), (\text{父亲, 孙子}), (\text{父亲, 孙女}), (\text{儿子, 女儿}), (\text{女儿, 孙子})\}$$

数据结构包括数据的逻辑结构和数据的物理结构。数据的逻辑结构可以看做是从具体问题抽象出来的数学模型,它与数据的存储无关。研究数据结构的目的是为了在计算机中实现对它的操作,为此还需要研究如何在计算机中表示一个数据结构。

数据结构在计算机中的表示(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,包括数据结构中元素的表示及元素间关系的表示。在计算机中表示信息的最小单位是“位(bit)”,任何一个数据元素都可以用一个“位串”表示。如,数值 321 可用位串 101000001 表示,字母 A 可用位串 001000001 表示。通常称这个位串为结点,当数据元素由多个数据项构成时,每个数据项即表示数据元素的位串中的一个“子位串”。

数据的存储结构可采用顺序存储或链式存储的方法。

顺序存储是把逻辑上相邻的元素存储在物理位置相邻的存储单元中,由此得到的存储表示称为顺序存储结构。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附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。

除了通常采用的顺序存储方法和链式存储方法外,有时为了查找的方便还采用索引存储和散列存储。