



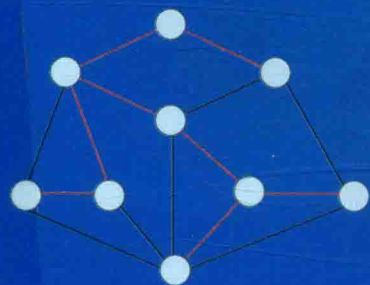
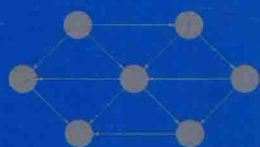
高等农林教育“十三五”规划教材

全国高等农业院校计算机类与电子信息类“十三五”规划教材

数据结构与算法

Data Structure and Algorithm

石玉强 闫大顺 主编



中国农业大学出版社

CHINA AGRICULTURAL UNIVERSITY PRESS



高等农林教育“十
全国高等农业院

“十三五”规划教材

数据结构与算法

石玉强 闫大顺 主编

中国农业大学出版社
• 北京 •

内 容 简 介

本书在选材与编排上,贴近当前普通高等院校“数据结构与算法”课程的现状和发展趋势,符合最新计算机专业研究生考试大纲,内容难度适中,突出数据结构的实用性和应用性。本书阐述了各种典型数据结构的基本概念、逻辑结构、存储结构以及相应各种操作。全书共8章,内容包括绪论、线性表、栈和队列、串、数组和广义表、树和二叉树、图、查找和排序等内容。每一章不仅有大量例题解析,还有丰富的习题。全书采用C/C++语言作为数据结构和算法的描述语言。本书可作为普通高等院校计算机和信息技术相关专业“数据结构”课程的教材使用,也可以作为报考高等学校计算机专业硕士研究生入学考试的复习用书,同时还可以作为从事计算机系统软件和应用软件设计与开发人员的参考资料。

图书在版编目(CIP)数据

数据结构与算法 / 石玉强, 闫大顺主编. —北京: 中国农业大学出版社, 2017. 1

ISBN 978-7-5655-1756-3

I. ①数… II. ①石…②闫… III. ①数据结构—高等学校—教材②算法分析—高等学校—教材 IV. ①TP311.12②TP301.6

中国版本图书馆 CIP 数据核字(2016)第 302433 号

书 名 数据结构与算法

作 者 石玉强 闫大顺 主编

策 划 司建新

责任编辑 冯雪梅

封面设计 郑 川

出版发行 中国农业大学出版社

社 址 北京市海淀区圆明园西路2号

邮政编码 100193

电 话 发行部 010-62818525, 8625

读者服务部 010-62732336

编辑部 010-62732617, 2618

出 版 部 010-62733440

网 址 <http://www.cau.edu.cn/caup>

E-mail cbsszs @ cau. edu. cn

经 销 新华书店

印 刷 涿州市星河印刷有限公司

版 次 2017年2月第1版 2017年2月第1次印刷

规 格 787×1092 16开本 18.75印张 460千字

定 价 48.00元

图书如有质量问题本社发行部负责调换

计算机类与电子信息类“十三五”规划教材 编写委员会

主 任 石玉强

副主任 闫大顺 冯大春

委 员 (以姓氏拼音为序)

杜淑琴	符志强	黄洪波	李 晟	罗慧慧	史婷婷
孙永新	王 潇	王俊红	吴志芳	肖明明	曾宪贵
张 垒	张世龙	赵爱芹	邹 莹		

前言

“数据结构与算法”是计算机程序设计的重要理论和实践基础,它不仅是计算机专业的核心课程,也是其他理工专业的重要选修课。在计算机的应用领域中,数据结构有着广泛的应用。

本书共分8章,第1章介绍了数据结构的基本概念和算法分析的初步知识;第2章到第4章介绍了线性表、栈和队列、串、数组和广义表等线性结构的基本概念及常用算法;第5章和第6章介绍了非线性结构的树、二叉树、图等数据结构的存储结构和不同存储结构上的一些操作的实现;第7章介绍了各种查找表及查找方法;第8章介绍了各种内存及外存排序算法。本书计划学时为80学时左右,其中上机实习为35学时左右。

本书是作者根据自己的教学经验总结,为计算机类普通高等院校应用型本科学生编写的教材。作者在教学过程中发现,大多数学生在初学数据结构时,经常误把算法的伪代码作为完整函数直接在编译器上进行运行测试。为了解决这个问题,本书采用C/C++语言描述数据结构和算法,并且对关键的算法都编写了完整的C语言程序供学生上机实习参考。书中给出的每一个算法都是完整的,只要添加变量定义和主函数,程序即可运行,主函数编写可以参考书中给出的案例程序,测试数据可以从需求分析获得。

应用型本科院校面向应用、注重实践,本书力求做到选材精练、叙述简洁、通俗易懂,尽量避免抽象理论的介绍和复杂公式的推导。对各种数据结构均从实际出发,通过对实例的分析,使学生理解数据结构的基本概念。

考虑到研究生入学考试和其他考试的需要,本书在每章后面带有适量的习题,并配有习题参考答案,方便学生自学参考。另外,与本书配套的多媒体教学课件、实验指导、习题参考答案均可从中国农业大学出版社网站下载,也可与作者联系,联系方式:yuqiangshi@163.com。

本书由石玉强、闫大顺任主编,孙永新、吴志芳、王俊红、曾宪贵、邹莹、王潇、张世龙任副主编。本书第1章由张世龙、史婷婷编写,第2章由石玉强编写,第3章由闫大顺编写,第4章由邹莹、孙永新、王潇编写,第5章由吴志芳编写,第6章由王俊红编写,第7章由顾春琴编写,第8章由曾宪贵编写,全书由石玉强、闫大顺统一编排定稿。

参加本书编写的还有刘磊安、杨灵、黄裕锋、符志强、李晟、冯大春、赵爱芹、罗慧慧、黄洪波、杜淑琴、刘佳、张垒、连剑波、郭世仁、陈勇、郑建华、贺超波、成筠、杨继臣、吴霆、杨现丽等,他们对书稿提出了宝贵的意见,在此一并表示衷心的感谢!

由于作者水平有限,书中难免会有不足和错误之处,敬请广大读者批评指正。

编者

2016年10月

目 录

第 1 章 绪论	1
1.1 数据结构的研究内容	1
1.2 基本概念和术语	4
1.2.1 数据、数据元素、数据项和数据对象	4
1.2.2 数据结构	5
1.2.3 数据类型和抽象数据类型	10
1.3 算法和算法分析	13
1.3.1 算法的定义及特性	13
1.3.2 算法评价的基本标准	14
1.3.3 算法的时间复杂度	15
1.3.4 算法的空间复杂度	17
小结	19
习题 1	19
第 2 章 线性表	21
2.1 线性表的类型定义	21
2.1.1 线性表的定义	22
2.1.2 线性表的抽象数据类型描述	22
2.2 线性表的顺序存储结构	26
2.2.1 线性表的顺序存储表示	27
2.2.2 顺序表中基本运算的实现	27
2.2.3 顺序表的特点	33
2.2.4 案例分析与实现	33
2.3 线性表的链式存储结构	34
2.3.1 线性表的链式存储表示	36
2.3.2 线性表基本运算在单链表中的实现	37
2.3.3 循环链表	43
2.3.4 双向链表	44
2.3.5 静态链表	48
2.3.6 案例分析	50
小结	55
习题 2	56

第3章 栈和队列	58
3.1 栈和队列的定义和特点	58
3.1.1 栈的定义和特点	58
3.1.2 队列的定义和特点	59
3.2 栈的表示和操作实现	60
3.2.1 栈的抽象类型定义	60
3.2.2 顺序栈的表示和实现	61
3.2.3 链栈的表示和实现	64
3.2.4 案例分析与实现	66
3.3 栈与递归	68
3.3.1 采用递归算法解决的问题	69
3.3.2 递归过程与递归工作栈	71
3.3.3 递归算法的效率分析	73
3.3.4 将递归转换为非递归的方法	73
3.4 队列的表示和操作实现	75
3.4.1 队列的抽象类型定义	75
3.4.2 循环队列——队列的顺序表示和实现	76
3.4.3 链队——队列的链式表示和实现	79
3.4.4 其他队列	83
3.5 典型栈和队列案例分析与实现	84
3.5.1 栈的典型用例	84
3.5.2 队列的典型用例	93
小结	95
习题3	96
第4章 串、数组和广义表	97
4.1 串的定义	97
4.1.1 串的定义	97
4.1.2 串的抽象数据类型描述	98
4.2 串的存储结构及其运算	99
4.2.1 串的定长顺序存储结构	100
4.2.2 串的堆分配存储结构	102
4.2.3 串的块链存储结构	105
4.2.4 串的模式匹配算法	107
4.3 数组	112
4.3.1 数组的类型定义	112
4.3.2 数组的顺序存储	113
4.3.3 特殊矩阵的压缩存储	114
4.4 广义表	120

4.4.1 广义表的定义	120
4.4.2 广义表的存储结构	121
4.5 案例分析与实现	123
小结	128
习题 4	128
第 5 章 树和二叉树	130
5.1 树	130
5.1.1 树的基本概念	130
5.1.2 树的抽象数据类型描述	133
5.1.3 树的遍历	134
5.1.4 树的存储结构	134
5.2 二叉树	136
5.2.1 二叉树的定义	136
5.2.2 二叉树的主要性质	137
5.2.3 二叉树的存储结构	139
5.2.4 树与二叉树之间的转换	142
5.2.5 二叉树的基本操作及实现	143
5.3 二叉树的遍历	149
5.3.1 二叉树的遍历方法及递归实现	149
5.3.2 二叉树遍历的非递归实现	151
5.4 线索二叉树	153
5.4.1 线索二叉树的概念	153
5.4.2 二叉树的线索化实现	155
5.4.3 线索二叉树上的运算	157
5.5 哈夫曼树及其应用	157
5.5.1 哈夫曼树概念	158
5.5.2 哈夫曼树构造算法	158
5.5.3 哈夫曼编码	161
5.6 案例分析——电文的编码和译码	163
小结	166
习题 5	167
第 6 章 图	168
6.1 图的定义和基本术语	168
6.1.1 图的定义	168
6.1.2 图的抽象数据类型描述	168
6.1.3 图的基本术语	169
6.2 图的存储结构	171
6.2.1 邻接矩阵	171

6.2.2	邻接表	172
6.2.3	十字链表	174
6.2.4	邻接多重表	174
6.2.5	图的存储结构设计举例	176
6.3	图的遍历	177
6.3.1	深度优先搜索	177
6.3.2	广度优先搜索	178
6.3.3	非连通图的遍历	179
6.3.4	图遍历算法的应用	179
6.4	图的应用	182
6.4.1	最小生成树	182
6.4.2	最短路径	188
6.4.3	拓扑排序	196
6.4.4	关键路径	198
6.5	案例分析与实现	201
小结		206
习题 6		207
第 7 章	查找	209
7.1	查找的基本概念	209
7.2	线性表的查找	210
7.2.1	顺序查找	211
7.2.2	折半查找	212
7.2.3	分块查找	214
7.3	树表的查找	215
7.3.1	二叉排序树	216
7.3.2	平衡二叉树	220
7.3.3	B-树	222
7.3.4	B+树	224
7.4	哈希表的查找	226
7.4.1	哈希表的基本概念	226
7.4.2	哈希函数的构造方法	227
7.4.3	处理冲突的方法	228
7.4.4	散列表的查找	230
小结		231
习题 7		232
第 8 章	排序	234
8.1	基本概念和排序方法概述	234
8.2	插入排序	235

8.2.1 直接插入排序	235
8.2.2 折半插入排序	237
8.2.3 希尔排序	238
8.3 交换排序	240
8.3.1 冒泡排序	240
8.3.2 快速排序	242
8.4 选择排序	245
8.4.1 简单选择排序	246
8.4.2 树形选择排序	247
8.4.3 堆排序	248
8.5 归并排序	253
8.6 基数排序	256
8.6.1 多关键字的排序	256
8.6.2 链式基数排序	259
8.7 外部排序	262
8.7.1 外部排序的基本方法	262
8.7.2 多路归并排序	263
8.7.3 置换-选择排序	265
小结	268
习题 8	269
附录 A C 语言常用语法提要	271
附录 B C 语言常用库函数	279
附录 C 实验报告模板	283
参考文献	286

第1章 绪论

学习要点

- 掌握数据结构的含义及其相关概念
- 掌握数据的常用的逻辑结构、存储结构
- 熟悉抽象数据类型的定义
- 熟悉算法的时间和空间复杂度的分析方法

随着计算机应用领域的不断扩大,非数值计算问题占据了当今计算机应用的绝大部分,数据元素之间的复杂联系已经不是普通数学方程式所能表达的了。掌握数据结构的知识及实践应用,将会提高解决实际问题的能力。实际上,一个“好”的程序无非是选择一个合理的数据结构和好的算法,而好的算法的选择很大程度上取决于描述实际问题所采用的数据结构,所以,要想编写出“好”的程序,仅仅学习计算机语言是不够的,必须扎实地掌握数据结构的基本知识和基本技能。

1.1 数据结构的研究内容

“数据结构”的概念最早是 C. A. R. Hoare 于 1966 年提出;在其《数据结构笔记》论文中,他首次提出了一组数据结构的构造、表示和操作等问题。1968 年 Donald. E. Knuth 教授在其所著《计算机程序设计艺术》第一卷《基本算法》中较为系统地阐述了数据的逻辑结构和存储结构及其操作,也开创了数据结构的课程体系。1976 年,结构化程序设计语言 PASCAL 之父、结构化程序设计的首创者、瑞士学者和计算机科学家 Niklaus Wirth 提出了著名公式:“程序 = 数据结构 + 算法”,表达了算法与数据结构的联系及其它们在程序中地位。

20 世纪 80 年代初我国正式开设了“数据结构”课程,目前“数据结构”已经成为介于数学、计算机硬件、计算机软件三者之间的一门核心课程。在计算机科学中,“数据结构”是一门综合性的专业基础课,也是计算机专业提高软件设计水平的一门关键性课程。“数据结构”的内容将为数据库原理、操作系统、编译原理、计算机原理、人工智能、数据挖掘、软件工程等后续课程的学习打下良好的基础。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。

通常用计算机解决一个具体问题需要如下步骤:首先从问题中抽象出一个适当的数学模型,再设计一个解该数学模型的算法,然后编写程序,进行测试、调整直至得到最终解答。计算机解决的问题可以概括为两类:一类是数值计算问题,指有效使用计算机求数学问题近似解的

方法与过程；一类是非数值计算问题，处理与自然界和人类社会的相关的文字、图形、图像、声音等数据。前者涉及的问题的数学模型能通过数学方程描述，操作对象一般是简单的整形、实型或布尔类型数据，无须重视操作对象之间的关系及存储；后者涉及的操作对象不再是简单的数据类型，其形式更多样、关系及结构更复杂，数学模型无法直接用数学方程进行问题及操作对象之间关系的描述。下面所列举的就是属于这一类的具体问题。

例 1.1 班级学生成绩表。

某一个计算机科学与技术专业的班级有程序设计基础、数据结构和操作系统等三门必修课，该班的成绩可用一个二维表格来表示，如表 1.1 所示。学号、姓名和课程名称是表头，表示了表格中各个数据的含义，每一行数据是一个学生的信息，通常称一行为一个记录。对学生成绩的处理是以每个学生为整体的，比如表 1.1 学生成绩按照学号排列，也可以按照总分从高到低的排序。像表格这样的数据，每行数据之间通常存在着的是一种最简单的前后关系，即根据某个学生的信息即可知道前面学生信息和后面的学生信息。类似于线性函数。

表 1.1 学生成绩单示例

学号	姓名	程序设计基础	数据结构	操作系统
20160101	徐成波	79	83	95
20160102	黄晓君	91	87	68
20160103	林宇珊	82	78	82
20160104	张茜	73	60	78
20160105	林宇珊	94	68	96
20160106	陈金燕	88	79	67
20160107	张顺峰	61	86	94
20160108	洪铭勇	77	96	67
20160109	朱伟东	86	62	78
20160110	叶剑峰	93	79	72
20160111	林宇珊	68	72	85
20160112	吴妍娴	97	82	87

诸如此类的线性表结构还有很多，例如图书馆的书目管理、超级市场的货品管理、通信录管理等。在这类问题中，计算机处理的对象是各种二维表，每行数据之间存在简单的一对一的线性关系，因此这类问题的数据模型被称为线性表，施加于对象上的操作有查找、插入、删除、更新等。这类数学模型称为线性的数据结构。

例 1.2 Linux 的文件系统。

Linux 是当今最流行的操作系统之一，是一个与 UNIX 兼容的操作系统。它起源于芬兰人 Linus Torvalds 于 1991 年赫尔辛基大学计算机系二年级在学习操作系统课程学习时编写完成的一个不完善的操作系统内核。区别于其他操作系统，Linus 把这个系统源代码放在 Internet 上，允许自由下载与修改，许多人对这个系统进行改进、扩充、完善，许多资深程序员和计算机公司做出了关键性贡献，使 Linux 成为广泛应用的操作系统。Linux 的文件系统是 Linux 操作系统的重要组成部分，通过目录来组织文件。Linux 文件系统只有一个根，形成一

个多级树型目录的形式,如图 1.1 所示,这样的结构称为树结构。

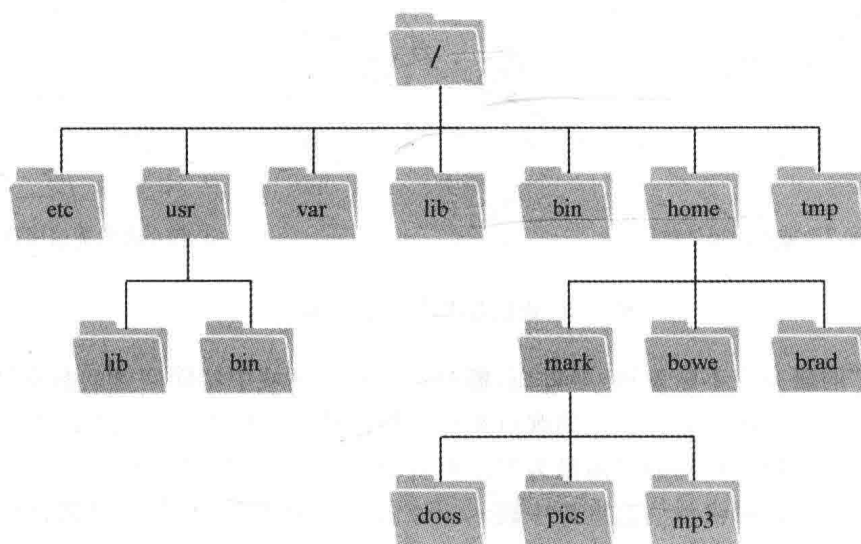


图 1.1 Linux 目录结构

诸如此类的树结构还有很多,例如一个单位的组织结构、族谱等。这类问题中,计算机处理的对象是树结构,树根或子树根与其分支结点是一对多,从而形成层次关系,施加于对象上的操作有查找、删除和插入等,但是这些操作都必须保持树形的结构不能破坏。其实线性结构是树结构的一种特殊情况,即树及所有的子树只有一个分支;但是树结构的操作有其特殊性,它不同于线性表的操作,比如人机对弈智能程序中有对弈树的裁剪、搜索树的形成等。这类数学模型称为树形的数据结构。

例 1.3 n 个城市间的通信网络。

在 n 个城市之间建立通信网络,要求在其中任意两个城市之间都有直接的或间接的通信线路。在已知城市之间直接通信线路建设预算造价的情况下,选择恰当的网络结构使网络的总造价最低。

当 n 很大时,这样的问题只能用计算机来求解。我们用图 1.2(a)来描述 7 个城市之间的通信线路,其中:图中圆圈表示一个城市,两个圆圈之间的连线表示对应城市之间的通信线路,连线上的数值表示该通信线路的造价,这样的结构称为图状结构,利用计算机可以求出满足要求的通信网络,如图 1.2(b)所示。

诸如此类的结构还有校园网拓扑结构图、工程建设项目图、铁路交通网、公路交通网等等,这些都是典型的网络结构,每个结点与多个其他结点互连,形成了元素之间的多对多的网状关系,图的操作依然为查找、插入和删除等,但它不同于树结构、线性结构的操作,比如最短路径求解、最短工期安排等等。这类数学模型称为图状的数据结构。

综上所述 3 个例子可见,这类非数值计算问题抽象出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。而解决这类非数值计算问题的算法与描述问题的数据结构密切相关,算法无不依附于具体的数据结构,不同数据结构的选择直接关系到算法的选择和效率。因此,为了编写出一个“好”的程序,必须分析待处理的对象的特征及各对象之间存在的关系,这就是数据结构这门课所要研究的问题。

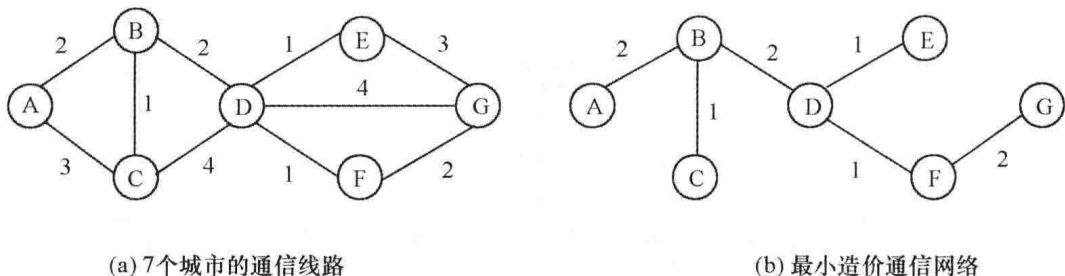


图 1.2 建立最小造价通信网络

“数据结构”的研究在不断发展,一方面,面向各个专门领域中特殊问题的数据结构正在研究和发展;另一方面,从抽象数据类型的观点来讨论数据结构,已经成为一种新的趋势,越来越被人们所重视。学习数据结构的目的是为了掌握计算机处理对象的特性,将现实世界的问题所涉及的处理对象在计算机的信息世界中表示出来并对它们进行处理,这是构造性思维能力的锻炼和提高,它将实现对程序抽象能力和数据抽象能力的强化。

1.2 基本概念和术语

“结构”是把某些成员按照一定的规律或方式组织在一起的实体;数据结构是以数据为成员的结构。

1.2.1 数据、数据元素、数据项和数据对象

1. 数据

数据(data)是对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能被计算机程序处理的符号的总称,它是计算机程序加工的“原料”。

例如,一个利用数值分析方法解代数方程的程序,其处理对象是整数和实数;一个编译程序或文字处理程序的处理对象是字符串。因此,对计算机科学而言,数据极为广泛,图形、图像、视频和音频等都可以通过编码而归之于数据的范畴。

2. 数据元素和数据项

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。一个数据元素可以是不可分割的原子(atom),例如一个整数;也可以由若干个数据项(data item)组成,例如一个记录,其中数据项是具有独立含义的最小数据单位,也称为字段或数据域。

例如,某程序处理的数据是学生情况登记表,每个学生的信息就是一个数据元素,其中的学生信息中的每一项(比如学号、姓名、性别、出生年月等)是这个数据元素中的数据项。图 1.3 所示即为上述数据元素的内部结构。

数据元素的同义词有:记录(record)、结点(node)和顶点(vertex)。它们的名称虽然不同,但所表示的意义却是一样的。通常,在顺序结构中多用“元素”,在链式结构中多用“结点”,在图和文件中又分别使用“顶点”和“记录”。

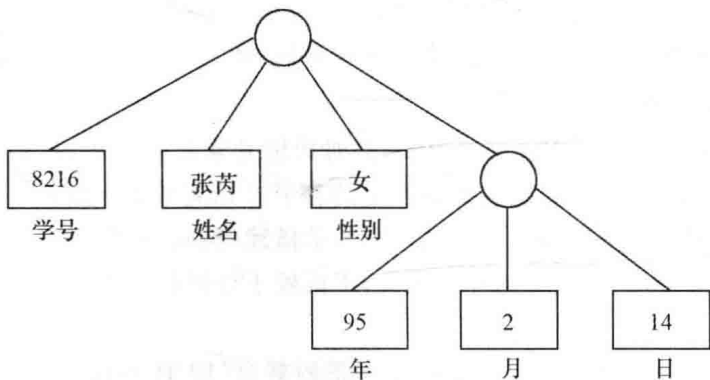


图 1.3 数据元素的内部结构

3. 数据对象

数据对象(data object)是性质相同的数据元素的集合,它是数据的一个子集。

例如,在电话号码查询系统中,数据对象是全体的电话用户;在银行业务处理系统中,数据对象是全体储户的资料及全体贷款客户的资料;学生的成绩表也是一个数据对象;整数就是所有整数的集合;单词是二十六个不同字母组成的集合。由此可见数据元素的集合无论是有限集合或是无限集合,只要集合内元素的性质相同,都可以称为一个数据对象。

4. 数据关系

在数据对象中各数据元素之间存在着某种关系,这种关系反映了数据对象中数据元素所固有的一种联系,这就是数据关系(data relation)。在数据处理领域,通常把数据之间这种固有的关系简单地用前驱和后继来描述。例如,在编写家庭族谱时,数据对象是家庭中的所有成员,对家族中某成员的描述就是一个数据元素,各数据元素之间存在着血缘关系;父亲是儿子的前驱,儿子是父亲的后继。由此,前驱和后继关系所表示的实际意义随着数据对象的不同而不同。在数据结构中,数据元素之间的任何关系都可以用前驱和后继关系来描述。

5. 关键字

关键字(关键码,key)指的是数据元素中能够起标识作用的那些数据项。其中能够起唯一标识作用的关键字称为“主关键字”(main key)。例如,在学生成绩表中,学号为关键字,姓名不能作为主关键字使用,因为会有学生重名的情况出现。

1.2.2 数据结构

对于数据结构这个概念,至今尚未有一个公认的定义,不同的人在使用这个词时所表达的意思有所不同。本书将给出一个通用的描述。

1. 数据结构

数据结构(data structure)是带有结构特性的数据元素的集合,它研究的是数据的逻辑结构和数据的物理结构以及它们之间的相互关系,并对这种结构定义相适应的运算,设计出相应的算法,并确保经过这些运算以后所得到的新结构仍保持原来的结构类型。简而言之,数据结构是相互之间存在一种或多种特定关系的数据元素的集合,即带“结构”的数据元素的集合。“结构”就是指数据元素之间存在的关系,分为逻辑结构和存储结构。

数据的逻辑结构和物理结构是数据结构的两个密切相关的方面,同一逻辑结构可以对应不同的存储结构。算法的设计取决于数据的逻辑结构,而算法的实现依赖于指定的存储结构。

2. 数据的逻辑结构

数据的逻辑结构(data logical structure)是对数据元素之间的逻辑(数学)关系的描述,它可以用一个数据元素的集合和定义在此集合上的若干二元关系来表示。

数据的逻辑结构定义是对操作对象的一种数学描述,换句话说,它是从具体问题的操作对象抽象出来的数学模型,它与数据的存储无关,不依赖于计算机。下面用数学方法给出数据的逻辑结构定义。

设 D 表示数据元素的集合, R 表示 D 上关系的集合(即 R 反映了 D 中各元素的前驱和后继关系,前驱或后继关系可以使用一个有序对表示 $\langle a_i, a_{i+1} \rangle$, 即 R 是有序对的集合),则一个数据的逻辑结构 B 可以表示为:

$$B = \{D, R\}$$

为了方便起见,常常用示意图表示数据的逻辑结构,其中小圆圈表示数据元素,关系用小圆圈之间的有向线段表示有序对,例如 $\langle a_i, a_{i+1} \rangle$ 为从 a_i 点指向 a_{i+1} 点的有向线;如果线段为两个方向,则省略方向为线段。根据数据元素之间的关系的不同特性,通常有 4 类基本结构。

(1) 集合(set): 在该数据结构中,只有数据元素,它们之间除了“同属于一个集合”外别无其他的关系,即 $R = \{\}$ 。集合是数据结构的一种特例,如图 1.4(a) 所示。

(2) 线性结构(line structure): 在该数据结构中,除了第 1 个数据元素外,其他各元素有唯一的前驱;除了最后一个数据元素外,其他各元素有唯一的后继。数据结构中的数据元素之间存在一对一的关系,如图 1.4(b) 所示。

(3) 树形结构(tree structure): 在该数据结构中,除了一个根数据元素(结点)外,其他各元素(结点)有唯一的前驱;所有数据元素(结点)都可以有多个后继。数据结构中的数据元素之间存在一对多的关系,如图 1.4(c) 所示。

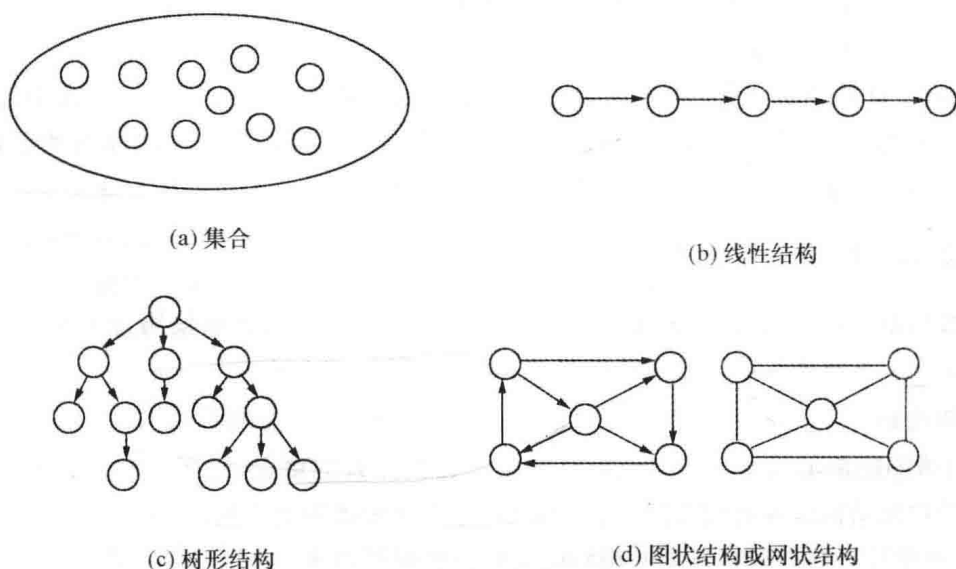


图 1.4 基本逻辑结构的示意图

(4)图状结构或网状结构(graph or met structure):在该数据结构中,各数据元素可以有多个前驱或后继。数据结构中的数据元素之间存在多对多的关系,如图 1.4(d)所示。

一般把树形结构和图状结构称为非线性结构。

在数据结构 $B = \{D, R\}$ 中,也可以不包含任何数据元素,即 $D = \{\}$,称为空数据结构。空数据结构到底属于哪种类型,应该视具体情况而定。

例 1.4 一周 7 天数据的逻辑结构。

在此结构中,有 7 个数据元素;有 1 个关系 R 。

$B = \{D, R\}$

$D = \{\text{Sun}, \text{Mon}, \text{Tue}, \text{Wed}, \text{Thu}, \text{Fri}, \text{Sat}\}$

$R = \{<\text{Sun}, \text{Mon}>, <\text{Mon}, \text{Tue}>, <\text{Tue}, \text{Wed}>, <\text{Wed}, \text{Thu}>, <\text{Thu}, \text{Fri}>, <\text{Fri}, \text{Sat}>\}$

以上数据的逻辑结构可以用图 1.5 形象地表示。

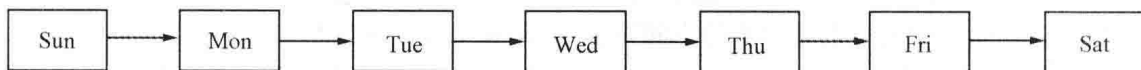


图 1.5 一周 7 天数据的逻辑结构

例 1.5 一种数据结构 $\text{Tree} = \{D, R\}$

$D = \{01, 03, 05, 07, 09, 11, 13, 15, 17, 19\}$

$R = \{<01, 03>, <01, 05>, <01, 07>, <03, 09>, <03, 11>, <03, 13>, <05, 15>, <05, 17>, <07, 19>\}$

这种数据结构的特点是除了结点“01”无直接前驱(称为根)以外,其余结点都只有一个直接前驱,但每个结点都可以有零个或多个直接后继,即结构的元素之间存在着 1 : N 的关系。

树形结构反映了结点元素之间的一种层次关系,如图 1.6 所示,从根结点起共分为三层,有向的箭头体现了结点之间的从属关系。

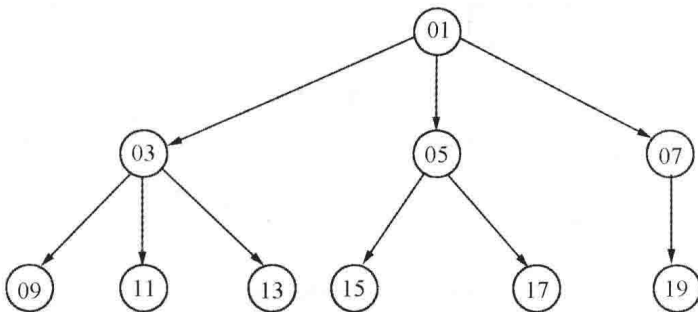


图 1.6 树形结构

例 1.6 一种数据结构 $\text{Graph} = \{D, R\}$

$D = \{1, 2, 3, 4, 5\}$

$R = \{(1, 2), (1, 4), (2, 4), (2, 3), (2, 5), (3, 4), (4, 5)\}$

圆括号表示的关系集合是无向的,如 $(1, 2)$ 表示从 1 到 2 之间的边是双向的。其特点是各