



普通高等教育“十一五”国家级规划教材

清华大学计算机系列教材

数据结构

(用面向对象方法与C++语言描述)
(第2版)

殷人昆 主编

清华大学出版社





普通高等教育“十一五”国家级规划教材

TP311.12C++
495(2)
12

清华大学计算机系列教材

数据结构

(用面向对象方法与C++语言描述)

(第2版)

殷人昆 主编

清华大学出版社
北京

内 容 简 介

“数据结构”是计算机专业的核心课程,是从事计算机软硬件开发和应用人员必备的专业基础。随着计算机的日益普及,“数据结构”课程也在不断地发展。

本书按照清华大学计算机系本科“数据结构”大纲的要求,从面向对象的概念、对象类设计的风格和数据结构的层次开始,从线性结构到非线性结构,从简单到复杂,深入地讨论了各种数据结构内在的逻辑关系及其在计算机中的实现方式和使用。此外,对常用的迭代、递归、回溯等算法设计技巧,搜索和排序算法等都做了详尽的描述,并引入了简单的算法分析。

全书采用面向对象的观点讨论数据结构技术,并以兼有面向过程和面向对象双重特色的 C++ 语言作为算法的描述工具,强化基本知识和基本能力的双基训练。全书条理清晰,通俗易懂,图文并茂,适于自学。

与本书配套的《数据结构习题解析——用面向对象方法与 C++ 语言描述》一书已经由清华大学出版社出版。本书适合大专院校计算机、软件专业本科生使用,也可作为教师和相关科研人员的参考书。

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 13501256678 13801310933

图书在版编目(CIP)数据

数据结构:用面向对象方法与 C++ 语言描述/殷人昆主编.——2 版.——北京:清华大学出版社,2007.6

(清华大学计算机系列教材)

ISBN 978-7-302-14811-1

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

中国版本图书馆 CIP 数据核字(2007)第 029518 号

责任编辑:马瑛珺 薛 阳

责任校对:时翠兰

责任印制:李红英

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

c-service@tup.tsinghua.edu.cn

社 总 机:010-62770175 邮购热线:010-62786544

投稿咨询:010-62772015 客户服务:010-62776969

印 刷 者:清华大学印刷厂

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185×260 印 张:33 字 数:799 千字

版 次:2007 年 6 月第 2 版 印 次:2007 年 9 月第 2 次印刷

印 数:5001~10000

定 价:39.00 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:010139-01

前 言

计算机的普及极大地改变了人们的工作和生活。目前各个行业、各个领域都与计算机建立了紧密的联系,也随之带来了开发各种软件的需求。为了能够以最少的成本,最快的速度,最好的质量开发出合乎需要的软件,必须遵循软件工程的原则,把软件的开发、维护标准化、工程化,不能再像以前那样,把软件看作是个人雕琢的精品。就软件产品而言,最重要的就是建立合理的软件体系结构和程序结构,设计有效的数据结构。因此,要做好软件开发工作,必须了解如何组织各种数据在计算机中的存储、传递和转换。这样,《数据结构》这门课程显得格外重要。自1978年美籍华裔学者冀中田在国内首开这门课程以来(当时作者也在场),经过20余年的发展,本课程已成为各大学计算机专业本科的主干课程,也成为非计算机类学生和研究生学习计算机的必修课程。

“数据结构”课程脱胎于“离散数学结构”,它涉及各种离散结构(如向量、集合、树、图、代数方程、多项式等)在计算机上如何存储和处理。其内容丰富,涉及面广泛,而且还在随各种基于计算机的应用技术的发展,不断增加新的内容。特别是面向对象技术出现以后,人们认识到,用它开发出来的软件体系结构更加符合人们的习惯,质量更容易得到保证,尤其是更容易适应使用者和用户不断提出的新的需求。因此,在国际上,面向对象技术得到迅速普及,出现了大批面向对象的软件开发工具。为了适合形势的要求,有必要开设结合面向对象技术的数据结构课程。

用面向对象的观点讨论数据结构,与传统的面向过程的讲法相比,变化较大。各种数据结构的讨论都是基于抽象数据类型和软件复用的,有新意,也有继承。我们力图与过去的讲授体系保持一致,但又必须引入一些新的概念。为了能够让读者容易学习,我们对内容进行了精选。许多从基本数据结构派生出来的概念,如双端堆、二项堆、最小-最大堆、斐波那契堆、左斜树、扁树、B*树等都舍去了。同时,把动态存储管理部分归到“操作系统”课程,把文件组织部分归到“数据库原理”,只保留了重要的应用最广泛的一些结构。对这些结构做全面深入的讲解,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,并结合各种典型事例说明它们在解决应用问题时的动态行为和各种必要的操作,并以C++语言为表述手段,介绍在面向对象程序设计过程中各种数据结构的表达和实现。只要是学过C或PASCAL语言,就能够很容易地阅读和理解,并因此学习C++语言,提高读者的软件设计和编程能力。

本书是作为清华大学信息学院平台课“数据结构”的教材编写的,在编写过程中得到清华大学信息学院领导的支持,并获得教育部“十一五”规划教材的资助。参与策划的有计算机系教师殷人昆、邓俊辉、舒继武、朱仲涛,电子系教师朱明方、吴及,自动化系教师李宛洲、刘义,微纳电子学系教师李树国,软件学院教师张力以及信息学院办公室的教师王娜等。第4章由舒继武执笔,第5章由朱仲涛执笔,第8章由邓俊辉执笔,第9章由吴及执笔,其他各章由殷人昆执笔。作者们都是在清华大学从事“数据结构”课程第一线教学的教师,有着丰富的数据结构和软件工程教学的经验,教学效果良好。

全书共分 10 章。第 1 章是预备知识,主要介绍什么是数据,数据与信息的关系;什么是数据结构,数据结构的分类。通过学习,读者能够了解抽象数据类型和面向对象的概念。并对对象、类、继承、消息以及其他关系的定义、使用有基本认识。由于我们选择了具有面向过程和面向对象双重特点的 C++ 语言,可以帮助读者自然而轻松地从传统程序设计观念向面向对象方法转变。在这一章的最后还讨论了算法设计和简单的算法分析方法。

第 2 章是全书的基础,讨论了线性表、它的数组表示和链表表示,以及利用它们定义出来的各种结构,如顺序表、代数多项式等。通过学习,读者可以了解对象和类的基本实现,并通过模板、多态性等的使用,对数据抽象概念有进一步的理解。

第 3 章引入 4 种存取受限的表,即栈、队列、优先级队列和双端队列。通过对它们的定义、实现和应用的深入介绍,使读者能够了解在什么场合使用它们,为以后更复杂的数据结构和算法的实现,提供了多种辅助手段。

第 4 章介绍在许多领域中经常遇到的多维数组、字符串和广义表。这些都是应用广泛又十分灵活的结构。

第 5 章和第 8 章介绍在实际应用中最重要非线性结构——树与图。在管理、电子设计、机械设计、日常生活中许多方面都会用到它们。

第 6 章、第 7 章和第 10 章介绍集合、跳表、散列、搜索树、索引以及文件等结构。在实际与信息处理相关的应用中,这些结构十分重要。许多非数值处理都涉及这些结构,它们与内存、外存上的数据组织关系密切。例如在外存组织文件时全面应用了这些结构。它们又是许多新结构的生长点。因此,读者学习这些内容将获益匪浅。

第 9 章介绍排序。这也是应用十分广泛的技术。只要是数据处理,就少不了排序。如何才能高效地完成排序,本章分别就内、外存使用的多种排序方法进行介绍和讨论,读者可以深入了解排序的机制,也能从中学到许多程序设计的技巧。

本书的篇幅虽然较大,但给读者以选择。可以根据时间、能力,适当对学习的内容加以剪裁。本着少讲多练的原则,可以对每种结构只介绍类定义和关键操作的实现,其他内容可自学。通过上机练习,加深理解。在本书目录中加**的章节可以酌情不讲。

在本书的成书过程中得到清华大学出版社的支持,在此表示衷心的感谢。最后需要说明的是,由于作者们的水平有限,书中肯定存在许多疏漏或错误,请读者及时来信批评指教。邮件地址是: yinrk@tsinghua.edu.cn 或 yinrk@sohu.com。

作 者

2006 年 11 月于北京

目 录

第 1 章 数据结构概论	1
1.1 数据结构的概念	1
1.1.1 数据结构举例.....	1
1.1.2 数据与数据结构.....	2
1.1.3 数据结构的分类.....	3
1.1.4 数据结构课程的内容.....	4
1.2 数据结构的抽象形式	6
1.2.1 数据类型.....	6
1.2.2 数据抽象与抽象数据类型.....	7
1.3 作为 ADT 的 C++ 类	9
1.3.1 面向对象的概念.....	9
1.3.2 C++ 中的类	10
1.3.3 C++ 中的对象	12
1.3.4 C++ 的输入输出	13
1.3.5 C++ 中的函数	14
1.3.6 动态存储分配	17
1.3.7 C++ 中的继承	18
1.3.8 多态性	19
1.3.9 C++ 的模板	23
1.4 算法定义.....	24
1.5 算法性能分析与度量.....	26
1.5.1 算法的性能标准	26
1.5.2 算法的后期测试	26
1.5.3 算法的事前估计	27
1.5.4 算法的渐进分析	32
**1.5.5 最坏、最好和平均情况	36
习题	37
第 2 章 线性表	43
2.1 线性表.....	43
2.1.1 线性表的概念	43
2.1.2 线性表的类定义	44
2.2 顺序表.....	45
2.2.1 顺序表的定义和特点	45
2.2.2 顺序表的类定义及其操作	45

2.2.3	顺序表的性能分析	50
2.2.4	顺序表的应用	52
2.3	单链表	52
2.3.1	单链表的概念	53
2.3.2	单链表的类定义	54
2.3.3	单链表中的插入与删除	56
2.3.4	带附加头结点的单链表	59
2.3.5	单链表的模板类	60
2.4	线性链表的其他变形	66
2.4.1	循环链表	66
2.4.2	双向链表	69
2.5	单链表的应用:多项式及其运算	73
**2.5.1	多项式的表示	74
**2.5.2	多项式的类定义	75
**2.5.3	多项式的加法	77
**2.5.4	多项式的乘法	79
2.6	静态链表	80
	习题	83
第3章	栈和队列	88
3.1	栈	88
3.1.1	栈的定义	88
3.1.2	顺序栈	89
3.1.3	链式栈	92
**3.1.4	栈的应用之一——括号匹配	94
**3.1.5	栈的应用之二——表达式的计算	95
3.2	栈与递归	101
3.2.1	递归的概念	101
3.2.2	递归过程与递归工作栈	105
**3.2.3	用回溯法求解迷宫问题	109
3.3	队列	114
3.3.1	队列的概念	114
3.3.2	循环队列	114
3.3.3	链式队列	118
3.3.4	队列应用举例:打印二项展开式 $(a+b)^i$ 的系数	120
**3.3.5	队列应用举例:电路布线	121
3.4	优先级队列	124
3.4.1	优先级队列的概念	124
**3.4.2	优先级队列的存储表示和实现	125

3.5	双端队列	126
3.5.1	双端队列的概念	126
3.5.2	双端队列的数组表示	128
3.5.3	双端队列的链表表示	129
	习题	131
第4章	数组、串与广义表	135
4.1	多维数组的概念与存储	135
4.1.1	多维数组的概念	135
4.1.2	多维数组的存储表示	136
4.2	特殊矩阵	138
4.2.1	对称矩阵的压缩存储	138
**4.2.2	三对角线/多对角线矩阵的压缩存储	140
4.3	稀疏矩阵	141
4.3.1	稀疏矩阵及其三元组数组表示	141
4.3.2	稀疏矩阵的转置	145
**4.3.3	稀疏矩阵的相加和相乘	147
**4.3.4	矩阵的正交链表表示	152
4.4	字符串	153
4.4.1	字符串的概念	153
4.4.2	C++ 有关字符串的库函数	154
4.4.3	字符串的实现	156
**4.4.4	字符串的自定义类	158
**4.4.5	字符串操作的实现	159
**4.4.6	字符串的模式匹配	161
**4.4.7	字符串的存储方法	167
4.5	广义表	169
4.5.1	广义表的定义与性质	169
4.5.2	广义表的表示	170
4.5.3	广义表存储结构的实现	170
**4.5.4	广义表的递归算法	174
**4.5.5	三元多项式的表示	181
	习题	183
第5章	树	186
5.1	树的基本概念	186
5.1.1	树的定义和术语	186
5.1.2	树的抽象数据类型	188
5.2	二叉树	189

5.2.1	二叉树的定义	189
5.2.2	二叉树的性质	190
5.2.3	二叉树的抽象数据类型	191
5.3	二叉树的存储表示	192
5.3.1	二叉树的数组存储表示	192
5.3.2	二叉树的链表存储表示	193
5.4	二叉树遍历及其应用	198
5.4.1	二叉树遍历的递归算法	198
5.4.2	二叉树遍历的应用	200
5.4.3	二叉树遍历的非递归算法	203
5.4.4	二叉树的计数	207
5.5	线索二叉树	210
5.5.1	线索	210
5.5.2	中序线索二叉树的建立和遍历	211
**5.5.3	中序线索二叉树的插入与删除	216
**5.5.4	前序与后序的线索化二叉树	218
5.6	树与森林	220
5.6.1	树的存储表示	220
5.6.2	森林与二叉树的转换	225
5.6.3	树与二叉树的转换	227
5.7	树与森林的遍历及其应用	227
5.7.1	树与森林的深度优先遍历	227
5.7.2	树和森林的广度优先遍历	230
**5.7.3	树遍历算法的应用	231
**5.7.4	其他基于遍历序列的几种存储表示	232
5.8	堆	235
5.8.1	最小堆和最大堆	235
5.8.2	堆的建立	236
5.8.3	堆的插入与删除	238
5.9	Huffman 树及其应用	240
5.9.1	路径长度	240
5.9.2	Huffman 树	241
**5.9.3	Huffman 树的应用:最优判定树	243
5.9.4	Huffman 树的应用:Huffman 编码	244
	习题	246

第 6 章 集合与字典 251

6.1 集合及其表示 251

6.1.1 集合的基本概念 251

6.1.2	用位向量实现集合抽象数据类型	252
6.1.3	用有序链表实现集合的抽象数据类型	257
6.2	并查集与等价类	262
6.2.1	并查集的定义及其实现	262
**6.2.2	并查集的应用:等价类划分	267
6.3	字典	268
6.3.1	字典的概念	269
6.3.2	字典的线性表描述	270
6.4	跳表	273
6.4.1	跳表的概念	273
**6.4.2	跳表的类定义	274
**6.4.3	跳表的搜索、插入和删除	276
6.5	散列	279
6.5.1	散列表与散列方法	279
6.5.2	散列函数	280
6.5.3	处理冲突的闭散列方法	282
6.5.4	处理冲突的开散列方法	291
6.5.5	散列表分析	293
	习题	294
第7章	搜索结构	297
7.1	静态搜索结构	298
7.1.1	静态搜索表	298
7.1.2	顺序搜索	300
7.1.3	基于有序顺序表的顺序搜索和折半搜索	302
**7.1.4	基于有序顺序表的其他搜索方法	307
7.2	二叉搜索树	308
7.2.1	二叉搜索树的概念	309
7.2.2	二叉搜索树上的搜索	310
7.2.3	二叉搜索树的插入	311
7.2.4	二叉搜索树的删除	313
7.2.5	二叉搜索树的性能分析	314
**7.2.6	最优二叉搜索树	317
7.3	AVL树	320
7.3.1	AVL树的概念	321
7.3.2	平衡化旋转	321
7.3.3	AVL树的插入	326
7.3.4	AVL树的删除	329
7.3.5	AVL树的性能分析	333

7.4	伸展树	334
7.5	红黑树	337
7.5.1	红黑树的概念和性质	337
7.5.2	红黑树的搜索	338
**7.5.3	红黑树的插入	338
**7.5.4	红黑树的删除	339
	习题	342
第8章	图	346
8.1	图的基本概念	346
8.1.1	与图有关的若干概念	346
8.1.2	图的抽象数据类型	348
8.2	图的存储结构	349
8.2.1	图的邻接矩阵表示	350
8.2.2	图的邻接表表示	355
**8.2.3	图的邻接多重表表示	361
8.3	图的遍历	363
8.3.1	深度优先搜索	364
8.3.2	广度优先搜索	365
8.3.3	连通分量	366
**8.3.4	重连通分量	368
8.4	最小生成树	370
8.4.1	Kruskal 算法	371
8.4.2	Prim 算法	373
8.5	最短路径	375
8.5.1	非负权值的单源最短路径	376
**8.5.2	任意权值的单源最短路径	379
**8.5.3	所有顶点之间的最短路径	381
8.6	用顶点表示活动的网络(AOV 网络)	383
8.7	用边表示活动的网络(AOE 网络)	388
	习题	392
第9章	排序	397
9.1	排序的概念及其算法性能分析	397
9.1.1	排序的概念	397
9.1.2	排序算法的性能评估	398
9.1.3	排序表的类定义	400
9.2	插入排序	401
9.2.1	直接插入排序	401

9.2.2	折半插入排序	403
9.2.3	希尔排序	404
9.3	快速排序	405
9.3.1	快速排序的过程	406
9.3.2	快速排序的性能分析	407
**9.3.3	快速排序的改进算法	409
**9.3.4	三路划分的快速排序算法	412
9.4	选择排序	413
9.4.1	直接选择排序	413
**9.4.2	锦标赛排序	414
9.4.3	堆排序	419
9.5	归并排序	422
9.5.1	归并	422
9.5.2	归并排序算法	423
9.6	基于链表的排序算法	425
**9.6.1	链表插入排序	425
**9.6.2	链表归并排序	427
**9.6.3	链表排序结果的重排	428
9.7	分配排序	431
9.7.1	桶式排序	431
9.7.2	基数排序	432
**9.7.3	MSD 基数排序	433
9.7.4	LSD 基数排序	435
9.8	内部排序算法的分析	437
9.8.1	排序方法的下界	437
9.8.2	各种内部排序方法的比较	439
	习题	440
第 10 章	文件、外部排序与搜索	444
10.1	主存储器和外存储器	444
10.1.1	磁带	444
10.1.2	磁盘存储器	446
10.1.3	缓冲区与缓冲池	448
10.2	文件组织	449
10.2.1	文件的概念	449
10.2.2	文件的存储结构	450
10.3	外排序	459
10.3.1	外排序的基本过程	459
10.3.2	k 路平衡归并与败者树	461

**10.3.3	初始归并段的生成(run generation)	466
**10.3.4	并行操作的缓冲区处理	470
**10.3.5	最佳归并树	473
10.4	多级索引结构	475
10.4.1	静态的 ISAM 方法	475
10.4.2	动态的 m 路搜索树	476
10.4.3	B 树	478
10.4.4	B 树的插入	480
10.4.5	B 树的删除	482
**10.4.6	B ⁺ 树	486
**10.4.7	VSAM	489
10.5	可扩充散列	490
**10.5.1	二叉 Trie 树	490
**10.5.2	将二叉 Trie 树转换为目录表	491
**10.5.3	目录表扩充与收缩	493
**10.5.4	性能分析	494
10.6	Trie 树	494
**10.6.1	Trie 树的定义	494
**10.6.2	Trie 树的搜索	495
**10.6.3	在 Trie 树上的插入和删除	496
	习题	497
附录 A 程序索引		500
附录 B 词汇索引		504
参考文献		512

第 1 章 数据结构概论

数据结构有用吗？当然有用，现在只要使用计算机，都会涉及数据结构，特别是从事计算机系统开发的专业人员，必须切实掌握数据结构的知识，它不仅是进入计算机行业的敲门砖，而且是开展计算机业务的技术基础。

数据结构难学吗？其实不难，只要仔细读书，认真思考，就能切实掌握各种结构的本质和它们之间的关系。有人说数据结构太难学，实际上是指动手编写程序难。这个问题是无法回避的，没有这方面的训练，将来就无法胜任系统开发的工作。动手能力的提高，需要日常无数次练习和总结。与学习任何一门课程一样，要踏踏实实地做题和实验，不断提高自己的编程能力和算法设计能力。

1.1 数据结构的概念

1.1.1 数据结构举例

众所周知，无论是使用计算机进行工程和科学计算，还是使用计算机做工业控制或信息管理，都属于数据处理的范畴。那么，如何在计算机中组织、存储、传递数据，就成为一个必须解决的问题。下面看几个例子。

【例 1】 在一个学生选课系统中，有两个数据实体（现实世界中的事物），即“学生”和“课程”，形成了两个数据表格。其中的“学生”实体包括了许多学生记录，它们按照每个学生学号递增的次序，顺序存放在“学生”表格中；而“课程”实体包括了各个课程记录，每个课程也有个课程编号，在“课程”表格中各个课程按照其课程编号递增的次序依次排列，如图 1.1 所示。

“学生”表格					“课程”表格				
	学号	姓名	性别	籍贯	出生年月	课程编号	课程名	学时	
1	98131	刘激扬	男	北京	1979.12	1	024002	程序设计基础	64
2	98164	衣春生	男	青岛	1979.07	2	024010	汇编语言	48
3	98165	卢声凯	男	天津	1981.02	3	024016	计算机原理	64
4	98182	袁秋慧	女	广州	1980.10	4	024020	数据结构	64
5	98203	林德康	男	上海	1980.05	5	024021	微机技术	64
6	98224	洪伟	男	太原	1981.01	6	024024	操作系统	48
7	98236	熊南燕	女	苏州	1980.03	7	024026	数据库原理	48
8	98297	宫力	男	北京	1981.01				
9	98310	蔡晓莉	女	昆明	1981.02				
10	98318	陈健	男	杭州	1979.12				

图 1.1 学生选课系统中的两个数据实体

在“学生”表格中,各个学生记录顺序排列,形成一个学生记录的线性序列,每个记录在序列中的位置有先后次序,它们之间形成一种线性关系。“课程”表格中的情况完全相同。

【例 2】 在学生选课系统中,一个学生可以选修多门课程,一门课程可以被多个学生选修,这在“学生”和“课程”实体之间形成多对多的关系。为了便于处理,引入一个新的实体“选课”,它的每一个记录是一张选课单,包含如下信息:(学号,课程编号,成绩,时间)。

此时,在“学生”实体、“课程”实体和“选课”实体间形成如图 1.2 所示的关系。这是一种网状关系。学生对选课、课程对选课都是一对多的关系。

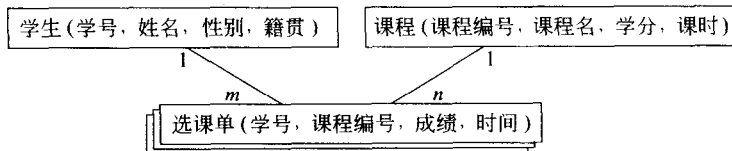


图 1.2 学生选课系统中实体构成的网状关系

【例 3】 一个典型的 UNIX 文件系统的系统结构如图 1.3 所示。这是一个层次结构:在此系统结构图中,顶层结点代表整个系统,用根目录/表示;它的下一层结点代表系统的各个子系统,即根的子目录,如/bin、/lib、/user 等;再下一层结点代表更小的子目录,如/user/yin、/bin/ds,如此类推,直到底层,即为文件,如/user/yin/queue.cpp。

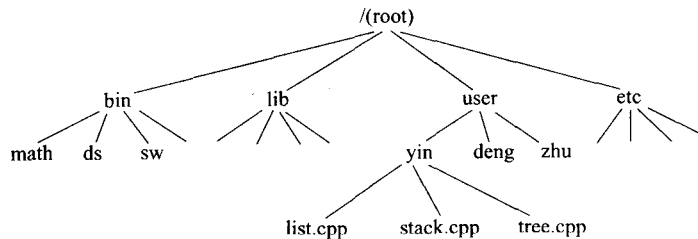


图 1.3 UNIX 文件系统的系统结构图

综上所述,在应用程序中涉及各种各样的数据,为了存储它们,组织它们,需要讨论它们的归类及它们之间的关系,从而建立相应的数据结构,并依此实现要求的软件功能。本课程的目的就是为系统开发者提供解决此问题的基本知识,使得读者在学习完本课程之后,能够在系统开发时运用学过的知识,在设计系统体系结构的同时设计出有效的数据结构,正确、高效地实现整个系统。

1.1.2 数据与数据结构

人们在日常生活中会遇到各种信息,如用语言交流的思想、银行与商店的商业交易、在战争中用于传递命令的旗语等。这些信息必须转换成数据才能在计算机中进行处理。因此,需要给数据下一个定义:

数据(data)是信息的载体,是描述客观事物的数、字符,以及所有能输入到计算机中并被计算机程序识别和处理的符号的集合。数据大致可分为两类,一类是数值性数据,包括整数、浮点数、复数、双精度数等,主要用于工程和科学计算,以及商业事务处理;另一类是非数值数据,主要包括字符和字符串,以及文字、图形、图像、语音等数据。

数据的基本单位是数据元素(data element)。一个数据元素可由若干个数据项(data item)组成,它是一个数据整体中相对独立的单位。例如,对于一个学籍管理文件来说,每个学生记录就是它的数据元素;对于一个字符串来说,每个字符就是它的数据元素;对于一个数组来说,每一个数组成分就是它的数据元素。

在数据元素中的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。

在数据处理中所涉及的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。例如,招生考试时把所有考生按考试成绩从高到低排队,所有考生记录都将处在一种有序的序列中;又例如,在 n 个网站之间建立通信网络,要求以最小的代价将 n 个网站连通,如图 1.4(a)所示,这样,在所有网站之间形成一种树形关系;反之,要求网络中任一网站出现故障,整个网络仍然保持畅通,这样,在所有网站之间形成一种网状关系,如图 1.4(b)所示。

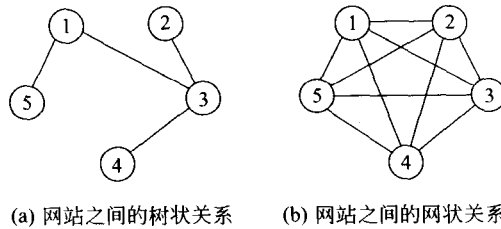


图 1.4 n 个网站之间的连通关系

由此可以引出数据结构的定义: **数据结构由某一数据元素的集合和该集合中数据元素之间的关系组成。**记为:

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

其中, D 是某一数据元素的集合, R 是该集合中所有数据元素之间的关系的有限集合。

注意,有关数据结构的讨论主要涉及数据元素之间的关系,不涉及数据元素本身的内容。关于数据元素的内容,在系统开发时考虑。

1.1.3 数据结构的分类

依据数据元素之间关系的不同,数据结构分为两大类:线性结构和非线性结构。

1. 线性结构(linear structure)

线性结构也称为线性表,在这种结构中所有数据元素都按某种次序排列在一个序列中,如图 1.5 所示。对于线性结构类中每一数据元素,除第一个元素外,其他每一个元素都有一个且仅有一个直接前驱,第一个数据元素没有前驱。除最后一个元素外,其他每一个元素都有一个且仅有一个直接后继,最后一个元素没有后继。

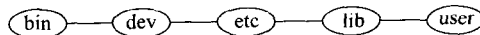


图 1.5 线性结构中各数据成员之间的线性关系

根据对线性结构中数据元素存取方法的不同,又可分为直接存取结构、顺序存取结构和字典结构。对于直接存取结构,可以直接存取某一指定项而不须先访问其先驱。像数组、文件都属于这一类。可以按给定下标直接存取数组中某一数组元素;可以按记录号直接检索记录集合或文件中的某一记录。对于顺序存取结构,只能从序列中第一个数据元素起,按序逐个访问直到指定的元素。像一些限制存取位置在表的一端或两端的表(如栈、队列和优先级队列等)就是这种情况。字典与数组有类似之处,但数组是通过整数下标进行索引,而字典是通过关键码(key)进行索引。我们设定数据元素中某一数据项或某一组合数据项为关键码,通过关键码来识别记录。例如,对于学生记录,可设定学生的学号为关键码,用它来识别是哪一位学生的记录。

2. 非线性结构(nonlinear structure)

在非线性结构中各个数据元素不再保持在一个线性序列中,每个数据元素可能与零个或多个其他数据元素发生联系。根据关系的不同,可分为层次结构和群结构。

层次结构(hierarchical structure)是按层次划分的数据元素的集合,指定层次上元素可以有零个或多个处于下一个层次上的直接所属下层元素。

树形结构就是典型的层次结构。树中的元素叫做结点。树可以为空,也可以不为空。若树不为空,它有一个叫做根的结点,其他结点都是从它派生出来的。除根以外,每一个结点都有一个处于该结点直接上层的结点。树的结构如图 1.6 所示。

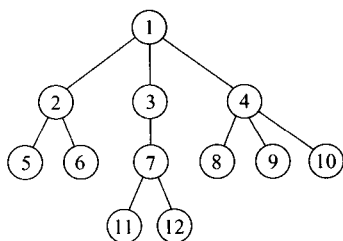


图 1.6 树形结构

注意,有的教科书规定树不能是空树,它至少必须有一个结点,而另外规定 k 叉树(即最多有 k 叉的有序树)可以是空树,并因此推定“ k 叉树不是树”。本书则对树与 k 叉树不加区分,在概念上将它们统一起来。

群结构(group structure)中所有元素之间无顺序关系。集合就是一种群结构,在集合中没有重复的元素。另一种群结构就是图结构,如图 1.7(a)所示。它是由图的顶点集合和连接顶点的边集组成。还有一种图的特殊形式,即网络结构。它给每条边赋予一个权值,这个权值指明了在遍历图时经过此边时的耗费。例如在图 1.7(b)中,顶点代表城市,赋予边的权值表示两个城市之间的距离。

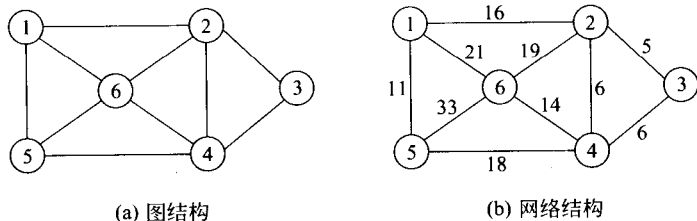


图 1.7 群结构

1.1.4 数据结构课程的内容

数据结构与数学、计算机硬件和软件有十分密切的关系。“数据结构”是介于数学、计算