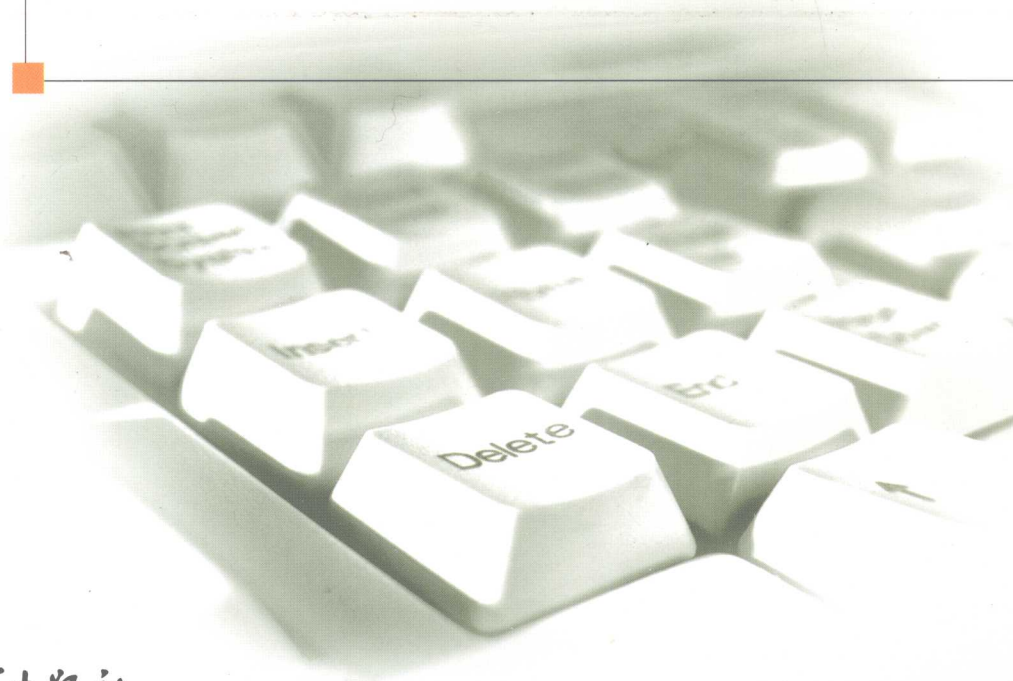
高职高专 计算机专业

GAOZHI GAOZHUAN
JISUANJI ZHUANYE XILIE JIAOCAI 系列教材

石曼银 主编

数据结构

(C语言描述)



厦门大学出版社
XIAMEN UNIVERSITY PRESS

数据结构(C语言描述)

主　编　石曼银

厦门大学出版社

图书在版编目(CIP)数据

数据结构:C语言描述/石曼银主编. —厦门:厦门大学出版社,2010.2
(高职高专计算机专业系列教材)
ISBN 978-7-5615-3482-3

Ⅰ.①数… Ⅱ.①石… 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099254号

厦门大学出版社出版发行
(地址:厦门市软件园二期望海路39号 邮编:361008)
http://www.xmupress.com
xmup@public.xm.fj.cn
南平市武夷美彩印中心印刷
2010年6月第1版 2010年6月第1次印刷
开本:787×1092 1/16 印张:15.75
字数:400千字 印数:1~3000元
定价:25.00元
本书如有印装质量问题请直接寄承印厂调换

前 言

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

本书是作者根据自己的教学经验总结，为高职高专计算机专业学生编写的教材。作者在教学过程中发现，大多数学生在初学数据结构时，容易误解算法与程序之间的关系，经常会把书中的算法当作程序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语言作为数据结构和算法的描述语言，并且对关键的算法都安排了完整的C语言程序供学生上机实习参考，程序均在VC编译器下编译运行。书中给出的每一个算法都是完整的，只要添加上主函数，程序即可运行，主函数的添加可以模仿书中给出的完整程序。

高职高专类院校面向应用，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。

考虑到专升本和其他考试的需要，在每章后面附有适量的习题。习题中编排了较多的选择题和填空题，并配有习题答案，方便学生自学参考。

本书共分9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5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的实现；第6章和第7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8章介绍了各种查找表及查找方法；第9章介绍了各种排序算法。本书计划学时为80学时左右，其中上机实习为35学时左右。

本书第1、5、6、7章由石曼银编写，第2、3、4章由张启宁编写，第8、9章由李志亮编写，全书由石曼银统一编排定稿。

本书的编写得到李豫颖副教授的指导；张丽芹、骆晓君、吴煌松、余业鑫、吴小英、夏芳芳等分别阅读了书稿中的部分章节，他们站在个人学习的角度对书稿提出了宝贵的意见，在此一并表示衷心的感谢！另外，衷心感谢厦门大学出版社

的编辑们,是他们的辛勤劳动,使本书得以顺利出版。

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

2010 年 6 月

目 录

前言
第 1 章 绪 论 …… 1
1.1 什么是数据结构 …… 1
1.2 数据结构的重要性 …… 3
1.3 基本概念和术语 …… 4
1.3.1 基本术语 …… 4
1.3.2 数据的逻辑结构 …… 5
1.3.3 数据的存储结构 …… 5
1.3.4 数据的运算与实现 …… 6
1.4 抽象数据类型 …… 7
1.5 算法与算法分析 …… 8
1.5.1 问题、算法和程序 …… 8
1.5.2 算法设计的要求 …… 9
1.5.3 算法分析 …… 10
本章小结 …… 12
练习题 …… 13
第 2 章 线性表 …… 16
2.1 线性表的基本概念 …… 16
2.1.1 线性表的自然语言定义 …… 17
2.1.2 线性表的 ADT 定义 …… 17
2.2 线性表的顺序存储结构及其运算 …… 18
2.2.1 顺序表的存储结构 …… 18
2.2.2 顺序表的基本操作 …… 19
2.2.3 顺序表的特点 …… 24
2.3 线性表的链式存储结构及其运算 …… 24
2.3.1 单链表的存储结构 …… 24
2.3.2 单链表的基本运算 …… 26
2.3.3 循环链表(Circular Linked List) …… 33
2.3.4 双向链表(Double Linked List) …… 33
2.3.5 线性表链式存储结构的特点 …… 35
2.4 线性表的应用举例 …… 35

2.5 上机实验 …… 37
2.5.1 实验目的 …… 37
2.5.2 实验内容 …… 37
本章小结 …… 38
练习题 …… 38
第3章 栈和队列 …… 41
3.1 栈的基本概念 …… 41
3.1.1 栈的自然语言定义 …… 41
3.1.2 栈的ADT定义 …… 42
3.2 栈的顺序存储结构及其运算 …… 42
3.2.1 栈的顺序存储结构 …… 42
3.2.2 顺序栈的基本操作 …… 43
3.3 栈的链式存储结构及其运算 …… 46
3.3.1 栈的链式存储结构 …… 46
3.3.2 链栈的基本操作 …… 46
3.4 栈的应用举例 …… 49
3.5 队列 …… 51
3.5.1 队列的自然语言定义 …… 51
3.5.2 队列的ADT定义 …… 51
3.6 队列的顺序存储结构及其运算 …… 52
3.6.1 队列的顺序存储结构 …… 52
3.6.2 循环队列 …… 53
3.6.3 循环队列的基本操作 …… 54
3.7 队列的链式存储结构及其运算 …… 57
3.7.1 队列的链式存储结构 …… 57
3.7.2 链队列的基本操作 …… 58
3.8 队列的应用举例 …… 61
3.9 上机实验 …… 62
3.9.1 实验目的 …… 62
3.9.2 实验内容 …… 63
本章小结 …… 63
练习题 …… 63
第4章 串 …… 66
4.1 串的基本概念 …… 66
4.1.1 串的自然语言定义 …… 66
4.1.2 串的ADT定义 …… 67
4.2 串的顺序存储结构及其运算 …… 68

4.2.1 串的顺序定长存储结构 …… 68
4.2.2 顺序串的基本操作 …… 68
4.3 串的堆分配存储结构及其运算 …… 70
4.3.1 串的堆分配存储结构 …… 70
4.3.2 串的堆分配存储结构的基本操作 …… 70
4.4 串的链式存储结构 …… 71
4.5 串的应用举例 …… 71
4.6 上机实验 …… 73
4.6.1 实验目的 …… 73
4.6.2 实验内容 …… 74
本章小结 …… 74
练习题 …… 74
第5章 数组和广义表 …… 76
5.1 数组的定义与存储 …… 76
5.1.1 数组的定义 …… 76
5.1.2 数组的顺序存储结构 …… 77
5.2 矩阵的压缩存储 …… 78
5.2.1 特殊矩阵 …… 78
5.2.2 稀疏矩阵 …… 81
5.3 广义表 …… 86
5.3.1 广义表的定义 …… 86
5.3.2 广义表的基本操作 …… 87
5.3.3 广义表的存储结构 …… 88
5.4 上机实验 …… 90
5.4.1 实验目的 …… 90
5.4.2 实验内容 …… 90
本章小结 …… 91
练习题 …… 91
第6章 树 …… 93
6.1 树的基本概念 …… 93
6.1.1 树的自然语言定义 …… 93
6.1.2 树的 ADT 定义 …… 95
6.1.3 树的表示方法 …… 95
6.1.4 树的基本术语 …… 96
6.2 树的存储结构 …… 97
6.2.1 树的顺序存储结构 …… 97
6.2.2 树的链式存储结构 …… 99

6.3 二叉树 …… 99
6.3.1 二叉树的定义 …… 100
6.3.2 二叉树的性质 …… 100
6.3.3 二叉树的存储结构 …… 102
6.3.4 二叉树的遍历 …… 105
6.3.5 线索二叉树 …… 109
6.3.6 二叉树的应用 …… 112
6.4 树、森林与二叉树之间的转换 …… 113
6.4.1 树、森林转换为二叉树 …… 114
6.4.2 二叉树转换为树、森林 …… 115
6.4.3 树和森林的遍历 …… 116
6.5 哈夫曼树 …… 117
6.5.1 哈夫曼树的基本概念 …… 117
6.5.2 构造哈夫曼树 …… 118
6.5.3 哈夫曼树的应用 …… 119
6.6 上机实验 …… 122
6.6.1 实验目的 …… 122
6.6.2 实验内容 …… 122
本章小结 …… 123
练习题 …… 123
第7章 图 …… 126
7.1 图的基本概念 …… 126
7.1.1 图的定义 …… 126
7.1.2 图的基本术语 …… 128
7.2 图的存储结构 …… 129
7.2.1 邻接矩阵 …… 130
7.2.2 邻接表 …… 133
7.3 图的遍历 …… 136
7.3.1 深度优先搜索遍历 …… 137
7.3.2 广度优先搜索遍历 …… 139
7.4 最小生成树 …… 140
7.4.1 生成树 …… 140
7.4.2 最小生成树 …… 141
7.5 最短路径 …… 150
7.5.1 从某个源点到其他各顶点的最短路径 …… 151
7.5.2 每对顶点之间的最短路径 …… 153
7.6 有向无环图及其应用 …… 156

7.6.1 拓扑排序 …… 156
7.6.2 关键路径 …… 162
7.7 上机实验 …… 165
7.7.1 实验目的 …… 165
7.7.2 实验内容 …… 165
本章小结 …… 165
练习题 …… 166
第8章 查 找 …… 170
8.1 查找的基本概念 …… 170
8.2 静态查找表 …… 171
8.2.1 顺序查找 …… 172
8.2.2 折半查找 …… 174
8.2.3 分块查找 …… 176
8.3 动态查找表 …… 177
8.3.1 二叉排序树 …… 177
8.3.2 平衡二叉树 …… 183
8.4 B－树和B＋树 …… 186
8.4.1 B－树 …… 186
8.4.2 B＋树 …… 189
8.5 哈希表及其查找 …… 189
8.5.1 哈希表的基本概念 …… 190
8.5.2 哈希函数的构造方法 …… 190
8.5.3 处理冲突的方法 …… 192
8.5.4 哈希表的平均查找长度 …… 194
8.6 上机实验 …… 195
8.6.1 实验目的 …… 195
8.6.2 实验内容 …… 196
本章小结 …… 196
练习题 …… 196
第9章 排 序 …… 199
9.1 排序的基本概念 …… 199
9.2 插入排序 …… 200
9.2.1 直接插入排序 …… 200
9.2.2 折半插入排序 …… 203
9.2.3 希尔排序 …… 205
9.3 交换排序 …… 207
9.3.1 冒泡排序 …… 207

9.3.2 快速排序 …… 209
9.4 选择排序 …… 211
9.4.1 直接选择排序 …… 212
9.4.2 堆排序 …… 213
9.5 归并排序 …… 216
9.6 各种排序方法的比较 …… 220
9.7 上机实验 …… 221
9.7.1 实验目的 …… 221
9.7.2 实验内容 …… 221
本章小结 …… 221
练习题 …… 222
参考答案 …… 224
参考文献 …… 239

第1章

绪 论

随着现代电子计算机技术的发展，计算机的应用领域从最初的简单科学计算逐步渗透到人类活动的各个领域，如工程过程控制、计算机辅助系统、人工智能、网络通信及管理信息系统等。计算机处理的数据已不再局限于数值数据，而是扩展到带有不同复杂结构的各种数据，这就给程序设计带来了一些新的问题。面对不同的数据处理对象和不同的需求，数据的组织形式、存储及运算必须有不同的方法，才能进行有效的处理。因此为了编写出好的程序，除了要掌握所用的计算机语言外，还需要研究各种数据的特性和数据之间存在的关系。这就是数据结构这门课程形成和发展的背景。

本章将介绍数据结构研究的对象、基本概念和术语、数据类型以及抽象数据类型，并介绍算法的概念及描述方法。

1.1 什么是数据结构

计算机是一种用来处理数据的设备。在解决实际问题时，一般是先对具体问题进行抽象，建立起实际问题求解模型，然后设计出相应的算法，最后编出程序并上机测试、调整直至得到最终解答，如图 1-1 所示。

图 1-1 计算机解决实际问题的过程

其中建立实际问题的求解模型的过程，实质是分析问题，从中提取操作对象，并找出这些操作对象之间含有的关系，然后用数学语言加以描述的过程。当处理的是数值计算问题时，所用的求解模型可以用数学方程来描述，所涉及的运算对象一般是整型、实型或逻辑型等一些简单的数据类型。程序设计者的主要精力集中在程序设计的技巧上，而不是数据的组织和存储。但是随着计算机应用领域的不断扩大，计算机处理的对象更多的是非数值计算问题，如图书资料查询、电话号码自动管理、交通道路规划、博弈游戏等问题，这些都无法用数学方程来加以描述，此时就必须建立相应的数据结构来进行描述，分析问题中用到的数据是如何组织的，研究数据之间的关系如何，进而为解决这些问题设计出合适的求解模型。请看下面 3 个例子。

【例 1.1】 图书管理问题。见表 1-1。

表 1-1 图书目录表

登录号	书号	书名	作者	出版社	定价
1	ISBN 7-302-02368-9/TP. 1185	数据结构	严蔚敏	清华大学	22
2	ISBN 7-302-00860-4/TP. 312	C 程序设计	谭浩强	清华大学	17.3
3	ISBN 7-5053-9279-4/TP. 311	数据结构	徐孝凯	电子工业	29
4	ISBN 7-5053-8168-7/TP. 4757	计算机系统原理	张基温	电子工业	25
5	ISBN 7-5609-2351-8/TP. 316	操作系统原理	庞丽萍	华中科技大学	22.8
6	ISBN 7-304-01404-0/TP. 68	数据库基础与应用	王 利	中央电大	23.3
7	ISBN 7-5084-1648-1/TP. 706	网页制作实例教程	齐建玲	中国水利水电	20

表 1-1 是一张图书目录表。在这个图书目录表中,每一行是一个**数据元素**(或称**记录**、**结点**),这张表表示一本书的信息。由于表中每条记录的登录号各不相同,所以可用登录号来唯一地标识每条记录(一本图书)。在计算机的数据管理中,能唯一地标识一条记录的数据项称为关键字。因为每本图书的登录排列位置有先后次序,所以在表中会按登录号形成一种次序关系,即整个二维表就是图书数据的一个线性序列。这种关系称为线性结构。

【例 1.2】 磁盘目录结构和文件管理系统。

图 1-2 是一个描述磁盘目录和文件结构的图。该磁盘目录结构中包括一个根目录(root)和若干个一级子目录,每个一级子目录中又包含若干个二级子目录……这种关系很像自然界中倒立的树,所以称为目录树。在这种结构中,目录和目录以及目录和文件之间呈现出一对多的非线性关系。即根 root 有多个下属(也称为后代),每个后代又有属于自己的后代,而任一个子目录或文件都只有一个唯一的上级(也称为双亲)。称这种数学模型为树形数据结构。

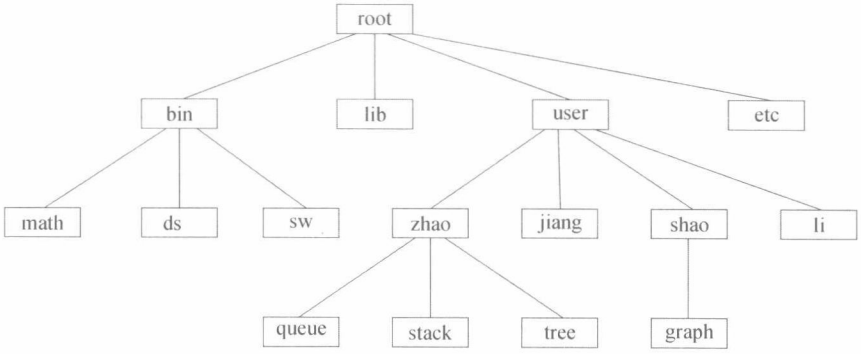


图 1-2 磁盘目录和文件结构

【例 1.3】 教学计划编排问题。

图 1-3 是表 1-2 所对应的修课图。通常一个教学计划中包含许多课程。在这些课程之间,有些必须按规定的先后次序排课,如学 C6 课程必须先学 C3 课程,学 C3 课程必须先学 C1 课程。这些课程之间存在先修和后续的关系,而且每一门课程的先修课程和后续课程都可能有若干门。在这种结构中,表示课程的数据之间呈现多对多的非线性关系,称这类数学模型为

图形结构。

表 1-2 教学课程编排表

课程编号	课程名称	先修课程
C1	计算机导论	无
C2	数据结构	C1,C4
C3	汇编语言	C1
C4	C 程序设计	C1
C5	计算机图形学	C2,C3,C4
C6	接口技术	C3
C7	数据库原理	C2,C9
C8	编译原理	C4
C9	操作系统	C2

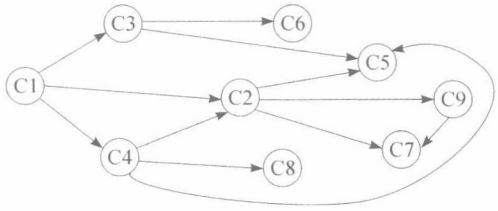


图 1-3 课程编排表对应的修课图

通过以上几例可以看出，描述这类非数值计算问题的数学模型不再是数学方程，而是诸如表、树和图之类的数据结构。这就是数据结构要研究的主要内容。

1.2 数据结构的重要性

数据结构和算法是程序设计最重要的两个内容。

数据结构作为一门独立的课程在国外是从 1968 年才开始设立的。1968 年美国唐·欧·克努特教授开创了数据结构的最初体系，他所著的《计算机程序设计技巧》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，人们就越来越重视数据结构，认为程序设计的实质是对确定的问题选择一种好的结构，加上设计一种好的算法，正如瑞士著名的计算机科学家 N. Wirth 提出的著名的公式“**程序＝算法＋数据结构**”所阐述的那样。从 70 年代中期到 80 年代初，各种版本的数据结构著作相继出现。

目前在我国，“数据结构”也已经不仅仅是计算机科学技术专业的教学计划中的核心课程之一，同时也是其他非计算机专业的主要选修课程之一。

“数据结构”在计算机科学中是一门综合性的专业基础课。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，而且和计算机软件的研究有着更密切的关系，无论是编译程序还是操作系统都涉及数据元素在存储器中的分配问题。在研究信息检索时也必须考虑如何组织数据，以便更方便查找和存取数据元素。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**。在计算机科学中，数据结构不仅是一般程序设计（特别是非数值计算的程序设计）的基础，而且是设计和实现编译

程序、操作系统、数据库系统及其他系统程序和大型应用程序的重要基础。

随着计算机科学技术的发展,计算机处理的数据量越来越大,数据的类型越来越多,数据的结构越来越复杂,仅仅掌握计算机语言和程序设计的技巧和方法,而缺乏有关数据结构的知识,几乎很难应付复杂的课题,更不能有效地利用计算机。

1.3 基本概念和术语

1.3.1 基本术语

数据(Data):是对客观事物的符号表示,是一种信息的载体。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。对计算机科学而言,数据的含义极为广泛,如整数、实数、字符、文字、图形、图像和声音等都是数据。

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。数据元素也可以称为记录、结点等,但它还可以分割成若干个具有不同属性的项(字段)。例如,在表 1-1 中,图书目录表中的一行就是一个数据元素。数据元素一般由一个或多个**数据项(Data Item)**组成。

数据项(Data Item):数据项是数据的不可分割的最小单位,是对数据元素属性的描述。在表 1-1 中,每个数据元素由 6 个数据项组成,分别为:登录号、书号、书名、作者、出版社和定价。

数据对象(Data Object):是性质相同的数据元素的集合。例如,在表 1-1 所示的图书目录表中,7 个数据元素的类型是相同的,我们可以把这 7 个数据元素看成是一个数据对象。数据对象由性质相同的数据元素组成,是数据的一个子集。

数据类型(Data Type):是指一组结构相同的值的集合以及定义于这个值的集合上的一组操作的总称。每个数据项属于某一确定的基本数据类型。例如表 1-1 中,登录号为整型,书号为字符型。数据类型有时还分为原子数据类型和结构数据类型。原子类型的值是不可分解的,如 C 语言中的基本类型:整型、实型、字符型等。结构类型的值由若干成分按某种结构组成,因此是可以分解的,如 C 语言中的构造类型:结构体、共用体、枚举等。

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。它是按照某种关系组织起来的一批数据,以一定的存储方式把它们存储到计算机存储器中,并在这些数据上定义了一个运算的集合。在任何问题中,数据元素都不是孤立存在的,在它们之间存在着某种关系,数据元素之间的这种相互关系就称为结构。一般地,数据结构包括以下三个方面的内容:数据的逻辑结构、数据的存储结构、数据的运算及实现。

1.3.2 数据的逻辑结构

数据元素之间的逻辑结构是从逻辑关系上描述数据，与数据的存储无关，是独立于计算机的。所谓逻辑关系是指数据元素之间的关联方式或称“邻接关系”。

如图 1-4(b)中用小圆圈表示数据元素，用连线表示元素之间的邻接关系，最左边第一个元素与第二个元素是邻接的(相互关联的)，因此这两个元素之间有逻辑关系。而第一个元素与第三个元素不邻接(相互之间不直接关联)，因此它们之间没有逻辑关系。

数据元素之间逻辑关系的整体称为逻辑结构。这里的“邻接关系”是一种抽象的关系，对于各种实际问题，它可以代表不同的关系。数据的逻辑结构主要分为两大类：**线性结构和非线性结构**，其中线性结构的逻辑特征是有且仅有一个开始元素和一个终止元素，除第一个和最后一个元素之外，其余元素有且仅有一个直接前驱和一个直接后继；非线性结构的逻辑特征是一个结点可能有零个或多个直接前驱或多个直接后继，包括集合结构、树形结构、图形结构(网状结构)。

(1)集合结构：数据元素之间除了同属于一个集合之外，没有其他关系的数据结构。

(2)线性结构：数据元素之间存在“一对一”逻辑关系的数据结构。

(3)树形结构：数据元素之间存在“一对多”逻辑关系的数据结构。

(4)图形结构(网状结构)：数据元素之间存在“多对多”逻辑关系的数据结构。

数据的四种基本逻辑结构如图 1-4 所示：

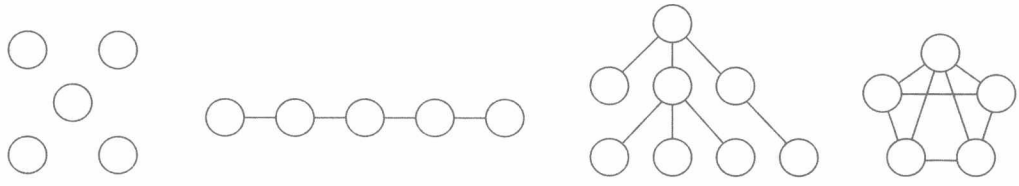


图 1-4 四种基本逻辑结构

1.3.3 数据的存储结构

数据的存储结构是数据的逻辑结构在计算机中的表示(又称映象)，它包括数据元素的机内表示和关系的机内表示。具体实现的方法有顺序存储结构、链式存储结构、索引存储结构、散列存储结构等多种，数据的具体操作与存储结构有很密切的联系。

(1)顺序存储结构：用一组地址连续的存储单元来表示数据元素之间的逻辑关系，结点间的逻辑关系由存储单元的相邻关系而体现，由此得到的存储表示称为顺序存储结构，通常可借助程序设计语言中的数组来描述，如图 1-5 所示。顺序存储结构的优点是占用较少的存储空间，可实现对各数据元素的随机访问。这是因为只要知道存储的首地址以及每个数据元素所占的存储单元，就可以计算出各数据元素的存储地址。缺点是由于只能使用相邻的一整块存

储单元,因此可能产生较多的碎片现象,并且不利于修改,在对数据元素进行插入和删除运算时可能要移动大量的数据元素,从而花费较多的时间。

图 1-5 顺序存储结构

(2)链式存储结构:不要求存储单元连续,数据元素可以存储在任意位置。为了实现数据元素之间逻辑关系的存储,将结点所占的存储单元分为两部分,一部分存放该结点本身的信息,另一部分存放该结点的后继结点地址,结点间的逻辑关系由附加的指针字段表示,如图 1-6 所示。链式存储结构常借助于程序设计语言中的指针类型描述。链式存储结构的优点是不会出现碎片现象,能充分利用所有的存储单元,并且在进行插入和删除操作时只需要修改指针,不需要移动大量元素,从而节省了时间;缺点是每个结点占用较多的存储空间。

图 1-6 链式存储结构

(3)索引存储结构:用结点的索引号来确定结点的存储地址。该方法通常是在存储结点信息的同时,建立附加的索引表。索引存储结构的优点是检索速度快;缺点是增加了附加的索引表,占用较多的存储空间,在插入和删除数据时需要修改索引表而花费较多的时间。

(4)散列存储结构:根据结点的关键字的值直接计算出该结点的存储地址,通过散列函数把结点间的逻辑关系对应到不同的物理空间。散列存储结构的优点是检索、插入和删除结点的操作都很快;缺点是当采用不好的散列函数时可能出现结点存储单元的冲突,为解决冲突需要附加时间和空间的开销。

一般来说,以上四种基本的存储方法,既可单独使用,也可以组合起来对数据结构进行存储。同一种逻辑结构采用不同的存储方法,可以得到不同的存储结构。例如,线性表若采用顺序存储方式,称为顺序表;若采用链式存储方式,称为链表;若采用散列存储方式,可称为散列表。选择何种存储结构来存储相应的逻辑结构要视具体问题的要求而定,也可依据运算是否方便及算法的时间效率和空间要求而定。

1.3.4 数据的运算与实现

数据的运算是指定义在数据的逻辑结构上的一组操作的集合。对数据操作的种类是没有限制的,可根据需要来定义,数据结构课程中讨论较多的操作有:检索(查找)、插入、删除、定位、修改、排序等。要注意的是,这些运算实际上是在逻辑结构上对抽象数据所施加的一系列抽象的操作,也就是说,对于这些操作,我们知道这些操作可以"做什么"而不需要考虑它们是"如何实现"的,只有对某种数据结构选定了存储结构之后,才去考虑如何将这些运算具体地实现,也就是运算的实现。**运算的实现依赖于所选取的存储结构,依赖于不同的计算机程序设计语言。**