

高等学校计算机教材

# 数据结构

陈向群 编著

人民邮电出版社  
[www.pptph.com.cn](http://www.pptph.com.cn)

高等学校计算机教材

# 数 据 结 构

陈向群 编著

人民邮电出版社

## 图书在版编目 (CIP) 数据

数据结构/陈向群编著.—北京:人民邮电出版社,2001.10  
高等学校计算机教材  
ISBN 7-115-09647-3

I.数... II.陈... III.数据结构—高等学校—教材 IV.TP311.12

中国版本图书馆 CIP 数据核字 (2001) 第 062727 号

## 内 容 提 要

本书介绍了数据结构的基本概念和基本算法以及数据结构中查找、排序的各种方法,并结合实例,给出了各种算法的 C 语言描述。

本书特点是从概念入手,注重分析各种数据结构算法的思路;着眼于应用,介绍了众多领域实际开发的实例。本书精选的 20 多个应用实例,涉及领域相当广泛,给读者提供了思路和方法,有助于提高读者分析和解决实际问题的能力。

本书可作为普通高等院校非计算机专业“数据结构”课程的教科书;对于计算机专业的学生或从事计算机应用的科技工作者,本书也是一本有价值的参考书籍。

高等学校计算机教材

## 数 据 结 构

◆ 编 著 陈向群  
责任编辑 滑 玉

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号  
邮编 100061 电子函件 315@pptph.com.cn  
网址 <http://www.pptph.com.cn>  
读者热线:010-67129212 010-67129211(传真)  
北京汉魂图文设计有限公司制作  
北京朝阳隆昌印刷厂印刷  
新华书店总店北京发行所经销

◆ 开本:787×1092 1/16  
印张:17.75  
字数:426 千字 2001 年 10 月第 1 版  
印数:1-5 000 册 2001 年 10 月北京第 1 次印刷

ISBN 7-115-09647-3/TP·2462

定价:24.00 元

本书如有印装质量问题,请与本社联系 电话:(010)67129223

## 编者的话

进入 21 世纪以来,我国的计算机应用与普及工作,以前所未有的规模在社会各个层面开展着,计算机已经成为自然科学、工程技术、管理科学和社会科学等等领域不可缺少的工具。

随着“数据结构”理论和方法在科学技术领域、工农业生产以及国民经济的各个部门的应用,作为计算机学科组成部分之一的“数据结构”也不再仅仅是高等学校中计算机相关专业的基础课程,而成为许多非计算机专业的必修课或选修课。

但是,不同学科和专业的基础理论和专业知识的学习要求毕竟是千差万别的。从人才培养和实际应用的角度考虑,不能把对计算机相关专业的学习要求原封不动地拿来要求非计算机专业的学生。为非计算机专业开设的“数据结构”课程,就应该与计算机相关专业的“数据结构”课程,从课程内容组织到学习重点安排等方面都有所区别,各有侧重。然而,真正能够从高等学校非计算机专业的需要出发而编写的“数据结构”教材则不多见。为此,我们编写了这本供高等学校各类非计算机专业使用的“数据结构”教材。

本书编写的指导思想是从概念入手,注重分析各种数据结构算法的思路;着眼于应用,广泛介绍众多领域实际开发的示例。

从上述指导思想出发,考虑到各非计算机专业的数学基础差别较大,本书对于数据结构算法的处理着重于深入浅出地讲述算法的思想和算法的基本操作步骤,避免抽象的数学描述、推理和证明。事实上,数据结构的各类基本算法都已经过严格的数学证明和大量实际应用的检验。对于多数读者来说,了解这些基本算法的思想以及考虑问题的角度,比知晓它们的抽象证明和推理更为重要。

同样,从上述考虑出发,书中列出的各类基本算法的 C 语言描述也是原理性的,并不是直接输入计算机就可执行的程序模块。这样安排的目的在于方便读者理解和掌握算法的实质,而不用拘泥于某一种开发工具的编程细节。实际上,用各种高级程序设计语言编写的数据结构的基本算法模块是非常容易找到的;当代软件开发工具在数据结构方面已经提供了大量的工具和函数,供开发人员使用。为此,本书专门安排了有关章节,介绍一些常见的开发工具对各种数据结构支持的内容。

为了帮助读者了解数据结构在实际工作的应用,本书通过各种渠道收集、精选了 20 多个与数据结构应用相关的实际例子。根据这些应用实例的各自特点,将它们分别安排在有关章节中。每一个例子都附有应用背景的介绍以及扼要的分析和评论。这些应用例子本身就给读者提供了解决实际问题的可能思路和方法,有助于提高读者分析实际问题和解决实际问题的能力。

本书精选的应用实例涉及的领域相当广泛,有关于机床行业的线切割编程控制软件设计的,有关于群落与生态系统的,有讨论多传感器侦察数据融合处理的,也有涉足风险评估与排序的,等等。本书中的大多数实际应用实例与数据结构在计算机领域自身的应用关系不大。这是本书与其他有关数据结构应用书籍的明显不同之处。

目前市面上已出版了一些数据结构习题集，其他一些数据结构的书籍中也附有不少数据结构的习题，因此，在本书每一章的后面，基本上不再附上这类习题，而代之的是一些以应用为主的练习题。最后，对参加本书资料收集、书稿编写、算法验证、插图绘制与内容审核工作的张乃琳、胡建钧、余啸海、杨达、吴立国、涂欣等人，表示由衷的谢意。

由于编者水平有限，书中难免有不妥或错误之处，恳请读者、专家批评指教。

编者

2001年7月

# 目 录

<b>第 1 章 概论</b> .....	1
1.1 学习数据结构的重要性 .....	1
1.2 什么是数据结构 .....	2
1.3 数据的逻辑结构 .....	3
1.3.1 基本概念.....	3
1.3.2 数据的逻辑结构.....	4
1.3.3 数据结构的分类.....	5
1.4 数据的存储结构 .....	6
1.5 数据运算和算法 .....	9
1.5.1 数据运算.....	9
1.5.2 算法.....	10
1.5.3 算法评价.....	11
习题.....	12
<b>第 2 章 线性表</b> .....	13
2.1 线性表的定义和运算 .....	13
2.1.1 线性表的定义.....	13
2.1.2 线性表的基本运算.....	14
2.2 线性表的顺序存储结构及其运算 .....	15
2.2.1 线性表的顺序存储结构.....	15
2.2.2 顺序表的运算.....	16
2.3 线性表的链接存储结构及其运算 .....	19
2.3.1 线性链表.....	19
2.3.2 单链表及其运算.....	21
2.3.3 双向链表.....	25
2.3.4 循环链表.....	28
2.4 线性表的应用举例 .....	30
2.4.1 在“线切割编程控制软件”中的链表 .....	30
2.4.2 在“多传感器侦察数据的融合处理”中的双向链表.....	34
习题.....	35
<b>第 3 章 堆栈与队列</b> .....	36
3.1 栈 .....	37

3.1.1	栈的定义	37
3.1.2	栈的基本运算	38
3.2	栈的存储结构与基本运算	38
3.2.1	栈的顺序存储结构	38
3.2.2	栈的运算	39
3.2.3	双栈结构	41
3.2.4	栈的链接存储结构及其运算	43
3.3	栈的应用举例	45
3.3.1	栈的应用之一: 表达式的计算	45
3.3.2	栈的应用之二: 数制转换	49
3.3.3	栈的应用之三: 括号匹配的检验	50
3.3.4	栈的应用之四: 微软 Word 软件中的“取消”与 “重复”命令的设计	52
3.4	递归	53
3.4.1	递归算法	53
3.4.2	递归的应用	54
3.4.3	递归算法分析	56
3.5	队列	56
3.6	队列的存储结构	57
3.6.1	队列的顺序存储结构	58
3.6.2	顺序队列的基本运算	59
3.6.3	循环队列	61
3.6.4	链接存储结构	63
3.7	队列的实际应用	65
3.7.1	在“打印机软件设计”中的队列	65
3.7.2	在“考虑冲突的模具生产计划调度系统”中的队列	65
3.7.3	在“智能排队系统”中的队列	69
	习题	71
<b>第4章</b>	<b>串</b>	<b>73</b>
4.1	串的定义	73
4.2	串的存储结构	74
4.2.1	顺序存储	75
4.2.2	链接存储	76
4.2.3	索引存储	77
4.3	串的基本运算	78
4.3.1	串的基本运算	78
4.3.2	串基本运算的 C 语言算法	78
4.4	串的应用举例	84

4.4.1 在“软件汉化”中的字符串 .....	84
4.4.2 在现代软件开发工具中的串操作 .....	84
习题 .....	86
<b>第5章 数组和广义表 .....</b>	<b>87</b>
5.1 数组 .....	88
5.1.1 数组的定义 .....	88
5.1.2 数组的顺序存储结构 .....	88
5.2 数组应用举例 .....	90
5.2.1 在“Visual Basic”中的数组 .....	90
5.2.2 Java 中的数组 .....	91
5.2.3 在“阵列处理机(数组处理机)”中的数组 .....	92
5.2.4 在“图形化编程语言 Lab VIEW”中的数组运算 .....	92
5.3 矩阵的压缩存储 .....	93
5.3.1 特殊矩阵的压缩存储 .....	93
5.3.2 稀疏矩阵及存储 .....	95
5.3.3 三元组表 .....	96
5.3.4 稀疏矩阵链接存储: 十字链表 .....	99
5.4 广义表 .....	100
5.4.1 广义表的定义 .....	100
5.4.2 广义表的存储结构 .....	101
5.5 特殊矩阵和广义表的应用举例 .....	103
5.5.1 在“群落与生态系统”研究中的三角矩阵 .....	103
5.5.2 在“基于 FMS 生产调度与控制的零件动态工艺模型” 中的稀疏矩阵 .....	105
5.5.3 在“中文字字同现概率统计及应用”中的稀疏矩阵 .....	107
习题 .....	108
<b>第6章 树和二叉树 .....</b>	<b>110</b>
6.1 树的基本概念 .....	110
6.1.1 树的定义 .....	110
6.1.2 树的表示方法 .....	112
6.1.3 树的存储结构 .....	113
6.2 二叉树 .....	114
6.2.1 二叉树的定义 .....	114
6.2.2 二叉树的基本性质 .....	115
6.3 二叉树的链接存储 .....	116
6.3.1 二叉链表 .....	116
6.3.2 二叉链表的生成 .....	117



6.4 二叉树的遍历 .....	119
6.4.1 二叉树遍历算法 .....	119
6.4.2 层次遍历算法 .....	122
6.4.3 遍历算法 .....	123
6.5 线索二叉树 .....	124
6.5.1 建立线索二叉树 .....	124
6.5.2 访问线索二叉树 .....	127
6.6 树、森林与二叉树的关系 .....	128
6.6.1 树与二叉树之间的转换 .....	128
6.6.2 森林与二叉树 .....	130
6.6.3 一般树和森林的运算 .....	131
6.7 哈夫曼树 .....	132
6.7.1 哈夫曼树的基本概念 .....	132
6.7.2 判定树 .....	134
6.7.3 哈夫曼编码 .....	135
6.8 树和二叉树的应用举例 .....	138
6.8.1 在“电力地理信息系统”中的树和二叉树的应用 .....	138
6.8.2 在“工程计算书自动生成技术”中的树的应用 .....	139
6.8.3 在“PLC 指令代码的文法分析和翻译”中的二叉树结构 .....	141
6.8.4 在“基于用户的 CAPP 零件编码系统的研究”中的二叉树 .....	142
习题 .....	145
<b>第 7 章 图</b> .....	<b>147</b>
7.1 图的基本概念 .....	147
7.2 图的存储结构 .....	151
7.2.1 邻接矩阵表示法 .....	151
7.2.2 邻接表表示法 .....	154
7.3 图的运算 .....	155
7.4 图的遍历 .....	157
7.4.1 深度优先搜索遍历 .....	157
7.4.2 广度优先搜索遍历 .....	160
7.4.3 无向图的遍历 .....	162
7.5 图的应用 .....	163
7.5.1 最小生成树 .....	163
7.5.2 最短路径 .....	168
7.5.3 拓扑 (Topology) 排序 .....	172
7.6 图的应用实例 .....	174
7.6.1 图在地理信息系统 (GIS) 中的作用 .....	174
7.6.2 在“变电站故障定位及恢复处理智能系统”中的图论应用 .....	178

---

7.6.3 在 Internet 路由器中的图论应用 .....	180
习题 .....	182
<b>第 8 章 查找</b> .....	<b>184</b>
8.1 基本概念 .....	184
8.2 线性表中的查找 .....	186
8.2.1 顺序查找 .....	186
8.2.2 二分法查找 .....	187
8.2.3 分块查找 .....	189
8.3 散列表及其查找 .....	192
8.3.1 散列表的概念 .....	192
8.3.2 散列函数的构造方法 .....	194
8.3.3 冲突处理 .....	196
8.4 查找的应用实例 .....	200
8.4.1 “口令”或“密码”的查找 .....	200
8.4.2 计算机病毒的检测技术 .....	201
8.4.3 在“电信通话记录”中的查找 .....	204
习题 .....	205
<b>第 9 章 排序</b> .....	<b>206</b>
9.1 排序的基本概念 .....	206
9.2 插入排序 .....	209
9.2.1 插入排序概述 .....	209
9.2.2 直接插入排序 .....	210
9.2.3 二分插入排序 .....	212
9.2.4 希尔排序 .....	214
9.3 选择排序 .....	216
9.3.1 直接选择排序 .....	216
9.3.2 树形选择排序 .....	218
9.3.3 堆排序 .....	220
9.4 交换排序 .....	226
9.4.1 冒泡排序 .....	226
9.4.2 快速排序 .....	229
9.5 归并排序 .....	232
9.5.1 二路归并 .....	232
9.5.2 一趟归并算法 .....	235
9.5.3 归并排序 .....	237
9.5.4 算法分析 .....	237
9.6 基数排序 .....	238

9.6.1 多关键字的排序 .....	238
9.6.2 基数排序 .....	239
9.7 几种排序方法的比较 .....	242
9.8 排序运算的实际应用事例 .....	244
9.8.1 在“全国普通高校招生网上录取系统”中的排序 .....	244
9.8.2 在“风险评估”中的排序 .....	246
习题 .....	249
<b>第 10 章 文件 .....</b>	<b>251</b>
10.1 文件的基本概念 .....	251
10.2 顺序文件 .....	253
10.3 索引文件 .....	254
10.3.1 索引顺序与 B <sup>+</sup> 树文件 .....	256
10.3.2 VSAM 文件 .....	259
10.4 散列文件 .....	260
10.5 多关键字文件 .....	262
10.5.1 多关键字文件的概念 .....	262
10.5.2 多重表文件 .....	263
10.5.3 倒排文件 .....	263
10.6 文件应用的实例 .....	264
10.6.1 在 Visual Basic 中的文件 .....	264
10.6.2 在“太湖流域水情自动监测系统”中的顺序文件 .....	266
10.6.3 在“超市商品库存”中的多关键字文件 .....	268
习题 .....	270
<b>参考文献 .....</b>	<b>272</b>

# 第 1 章 概 论

本章要点:

- 数据结构的基本概念
- 数据的逻辑结构及分类
- 数据的存储结构
- 数据的运算
- 算法的概念、分析和评价

数据结构的基本概念是学习数据结构的基础。因此有必要掌握数据结构的基本概念,包括数据、数据元素、数据结构、逻辑结构、存储结构、数据运算和算法设计等。

## 1.1 学习数据结构的重要性

从原理上讲,计算机是一种自动处理数据的装置。在数据输入计算机之后,计算机对数据进行处理,得到所需要的数据结果。因此,数据是计算机处理的对象,而对数据处理过程的抽象描述则被称为算法。

早期的计算机价格昂贵,运算速度慢,存储容量也有限,主要用于科学计算。其特点是计算量大,但数据量少,相应的数据结构也简单。所以,当时人们并不重视对数据结构的研究。

随着计算机技术的发展,计算机大量应用于非数值计算领域,像银行业务处理、工商企业管理和政府部门管理等。其应用特点是需要处理大量数据,但数据处理过程相对比较简单。比如银行业务的处理,需要输入、存储大量存款客户和贷款客户的详细资料,对这些数据的处理无非是记录、分类、汇总、查找、计算利息等简单工作。面对大批量要处理的数据,为了有效地进行处理,就必须研究数据自身的内在结构。

随着各种高级程序语言的相继出现,这些高级语言所能描述的数据类型也逐渐丰富,例如在 FORTRAN 语言中出现了复数类型和数组类型,在 COBOL 语言和 PL/1 语言中允许使用字符类型和记录类型,而数组类型和记录类型都已具备“结构”的雏型。

20 世纪 60 年代中期,美国计算机界首先使用了数据结构这一名称。1968 年美国著名计算机专家唐·欧·克努特教授创立了数据结构的最初体系,他在《计算机程序设计技巧》第一卷《基本算法》一书中,全面、系统地阐述了数据的逻辑结构和存储结构,并给出了各种典型算法,为数据结构奠定了基础。

## 1.2 什么是数据结构

数据结构和算法是程序设计中两个同等重要的问题，二者相辅相成，缺一不可。

那么，什么是数据结构呢？简而言之，数据结构是指数据之间的关系。

在计算机处理数据过程中，大批量的数据之间不是彼此孤立，杂乱无章的，它们之间有着内在的联系。通过利用数据内部客观存在的关系，可以把各数据元素有机地组织起来，从而有可能对这些数据进行有效的处理。

我们以银行中的数据为例，来说明这个问题。

首先，引入记录的概念。记录，是把一组相关联的不同类型的数据看作一个整体。这样，一个银行储户的基本情况，包括姓名、地址、账号、身份证号、开户日期、存款余额等项目的内容，就可以组织起来，构成一个储户的基本资料，称为该储户的记录。再把各储户的记录组合起来构成银行全部储户的数据。当我们要找一个储户的基本资料时，不需要分别去找该储户的姓名、地址、账号、身份证号、开户日期、存款余额等信息，只要让计算机调出该储户的记录，储户所有的基本资料就一目了然。这就是数据结构的作用。

下面我们考虑储户记录的查找问题。

银行储户的名册是在经常变化的。查找储户的最主要的工作是，当给出储户的个人名字或账号时，能在储户记录中迅速查找到其账号中的有关信息，如还有多少存款，存期多长，利率是多少等。

当然，当有新储户要加入或旧储户要提取存款、增加存款或撤消账号时，也要对储户名册进行相应修改。那么应该怎样组织银行储户名册，使得以上操作能够实现呢？最简单的组织方式如图 1-1 所示，把各储户名册随机罗列出来。这样，要查找储户的基本资料，只有从表中第一行开始，依次往后顺序查找，只要被查找者确实有账号在册，就会找到。同样对旧储户记录的修改、删除也可以用类似方式操作。而对于新储户记录的添加，则可以添加在原有储户名册的末尾。但是随机罗列储户名册，然后采用顺序查找的方式，效率必然是很低的。显然，随机罗列储户名册不是一个好的方法。

储户名	账号	存款(元)	存期(月)	利率(%)	起始日期	备注
蔡新园	90030145	30000.00	12	5.5000	1999.6.23	
丛雨桦	80020234	1200.00	6	5.0625	2000.1.20	
戴沙兵	70010567	34000.00	12	5.5000	1999.3.05	
丁温慧	90040432	56000.00	24	5.6250	2001.6.06	
唐槐鸿	70054230	67000.00	12	5.5000	1999.11.28	
黄席华	80050231	1000.00	24	5.6250	1999.01.15	
樊伞标	90045893	5700.00	12	5.5000	2000.07.23	
孔尖湘	90070879	320000.00	6	5.0625	2000.05.14	
范佩佩	70056980	4500.00	24	5.6250	2001.01.12	
鲍源恺	60034435	3200.00	12	5.5000	2000.02.04	
皇钊茹	90020943	10000.00	12	5.5000	2000.07.16	

图 1-1 银行储户的名册

为了提高查找效率，我们可以这样组织储户名册，即让储户按姓氏笔划顺序登录，并为不同的姓氏按笔划各自建立一张索引表。假设要查找丁温慧储户的基本资料，先查找姓氏笔划检索表，找到丁姓索引表的位置，再在丁姓索引表中查找温慧的名字，这样，每次查找范围缩小了，整体查找速度就加快了。

因此，数据组织方式不同，进行同样的内容查找工作，其查找算法各不相同，查找效率当然也不同。

要在计算机上查找银行储户名册，当然首先要把以上数据存储到计算机中。用什么方式存储，使之能体现出数据之间的关系，使得数据处理能正确、高效完成，这也是数据结构应讨论的问题。

因此，数据结构讨论的问题应包括数据的逻辑结构、数据的存储结构及数据运算的表示三个方面。

把银行储户名册以记录的方式组织，并存储到计算机中，这就是一个数据结构。

而银行储户名册表是一个最简单、最常用的数据结构，称为线性表。

线性表的逻辑结构应能解决哪一个数据元素是表中的第一个元素，哪一个元素是表中的最后一个元素，在某个指定元素的前一个元素是哪一个元素，在某个指定元素的后一个元素又是哪一个元素等问题。

线性表的存储结构应能说明线性表中各元素在存储器中是顺序存储的（用存储单元地址的连续性体现线性表的逻辑关系），还是链接存储的（用指针链接体现数据元素间的逻辑关系）。

线性表的数据运算包括：在表中查找一个元素，在表中插入一个元素或在表中删除一个元素等操作。

综上所述，我们认识到，以某种关系组织起来的一批数据，用一定的存储方式存储到计算机的存储器中，并且在这些数据上定义了一个运算集合，这就是数据结构。

## 1.3 数据的逻辑结构

### 1.3.1 基本概念

**数据** 指能输入计算机且能被计算机处理的一切对象。如整数、实数、字符、文字、逻辑值、图形、图像、声音等都是数据。数据是信息的载体，是对客观事物的描述。

**数据元素** 是对现实世界中某独立个体的数据描述。数据元素是数据的基本单位。如前面列举的一个银行储户资料就是一个数据元素。数据元素一般由一个或多个数据项组成。数据元素有时也称为结点、记录或表目。

**数据项** 是具有独立意义的最小数据单位。数据项是对数据元素属性的描述。数据项也称域或字段。如前面列举的一个银行储户资料记录中，每个储户数据元素由多个数据项组成，其中“储户名”数据项描述了储户的名字，“账号”数据项描述了该储户的账号号码。

**数据类型** 每个数据项属于某个确定的基本数据类型。如银行储户资料中，“储户名”是字符串型，“起始日期”是日期型，“存款”是实数型等。一个数据元素如果只包含一个数

据项，则数据项的类型就是该数据元素的类型，否则数据元素的类型由各种数据类型构造出来。

**数据对象** 指具有相同特征的数据元素的集合。例如在银行业务处理中，数据对象就是全体储户资料及全体贷款客户的资料。

**关系** 是在数据对象中各数据元素所固有的一种结构。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、后继关系来描述。例如在编写一家人的家族谱时，数据对象是这家人的所有成员，对家族中各个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，小孩子没有后继。又如考虑一张按名次排列的成绩表，其数据对象是全班同学，对某个同学属性（姓名、成绩等）的描述就是一个数据元素，各数据元素之间存在着名次关系。第一名没有前驱，其后继是第二名；第二名的前驱是第一名，其后继是第三名。因此，前驱后继关系所表示的实际意义随着数据对象的不同而不同。一般地，数据元素之间的任何关系都可用前驱、后继关系描述。

某个结点（数据元素）如果没有前驱，则该结点为根结点或起始结点；某个结点如果没有后继，则称此结点为终端结点。

### 1.3.2 数据的逻辑结构

#### 1. 逻辑结构的定义

本节用数学方法给出逻辑结构的定义。

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、后继关系），则一个数据结构  $B$  可以表示为：

$$B = (D, R)$$

因此，数据结构由两部分构成：

- (1) 表示各数据元素的信息  $D$ ；
- (2) 表示各数据元素之间关系的信息  $R$ 。

在数据结构领域中，一般用一个二元组表示  $D$  中各数据元素之间的前驱、后继关系。例如，假设  $a, b$  是  $D$  中的两个数据元素，则二元组  $\langle a, b \rangle$  表示  $a$  是  $b$  的前驱， $b$  是  $a$  的后继。

#### 2. 数据结构的例子

一周七天的数据结构可表示为：

$$B = (D, R)$$

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

$$R = \{\langle \text{Sun, Mon} \rangle, \langle \text{Mon, Tue} \rangle, \langle \text{Tue, Wed} \rangle, \langle \text{Wed, Thu} \rangle, \langle \text{Thu, Fri} \rangle, \langle \text{Fri, Sat} \rangle\}$$

本例的数据结构可用图 1-2 表示。

图 1-2 称为数据的图形表示，图中方框表示一个结点，框内文字是该结点的值，带箭头的线段表示前驱、后继关系。

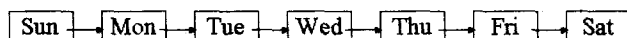


图 1-2 一周七天的数据结构

### 3. 若干约定

为了便于读者的阅读，对有关约定如下：

(1) 数据元素，也称结点、记录、表目、顶点等，以后统称为结点。

(2) 实际上，结点都是由若干个数据项组成的，即结点类型是一个结构体类型。考虑到我们在讨论数据结构时，研究重点是结点之间的关系，并不涉及结点的内部结构，因此，如果没有特别说明，我们都假设结点的类型为整型 (*int*)。

(3) 通常，一个数据结构存在着若干个关系。为方便讨论起见，我们假设在一个数据结构中只存在一个关系。实际上，如果一个数据结构包含有多个关系，可以用类似的方法分别讨论。

(4) 在不引起混乱的前提下，把数据的逻辑结构简称为数据结构，把数据的存储结构简称为存储结构。

### 1.3.3 数据结构的分类

我们把数据结构分为四种基本类型，其分类依据是按照数据结构中各结点之间前驱、后继结点的数目。

**集合** 在这种数据结构中只有结点，结点之间不存在关系，即  $R = \{\}$ ，如图 1-3 (a) 所示。这是数据结构的一种特例，本书不予讨论。

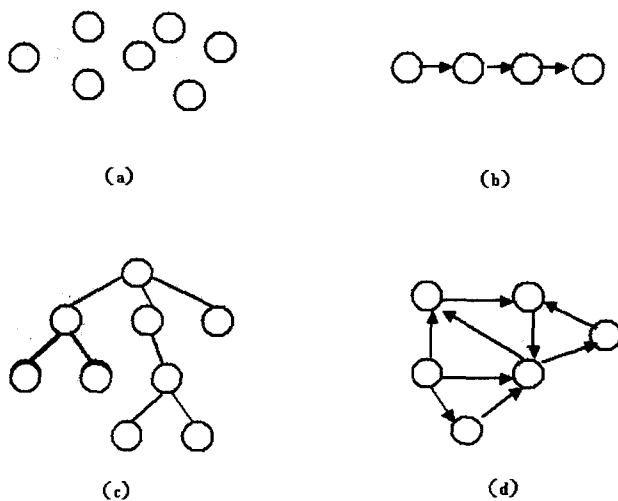


图 1-3 四种数据结构基本类型

**线性结构** 在该数据结构中，除了一个根结点外，其他各结点有唯一的前驱；除一个



终端结点外，其他各结点有唯一的后继。这是一种一对一的关系，如图 1-3 (b) 所示。

**树型结构** 在该数据结构中，除了一个根结点外，各结点有唯一前驱；但是所有结点都可以有多个后继，这是一种一对多的关系，如图 1-3 (c) 所示。

**图状或网状结构** 在该数据结构中，各结点可以有多个前驱或多个后继。这是一种多对多的关系，如图 1-3 (d) 所示。

通常，树型结构和图状结构被称为非线性结构。

在数据结构  $B = (D, R)$  中，也可以不包含任何结点，即  $D = \{ \}$ ，这称为空数据结构。至于空数据结构到底属于哪种类型，应视具体情况而定。

## 1.4 数据的存储结构

数据结构在计算机中的表示称为数据的存储结构，简称存储结构。

怎样既能合理利用有限的存储空间，又能有效地处理数据结构，这是存储结构要讨论的问题。

计算机存储器（内存）是由很多存储单元（word 或 byte）组成的，每个存储单元都有唯一的地址。各存储单元的地址编码是线性连续的，即每一个存储单元有唯一的前驱单元和唯一的后继单元。我们把两个互为前驱后继的存储单元称为相邻存储单元，把一片相邻的存储单元称为存储区域。

一般情况下，一个结点所占用的空间并不只一个存储单元，不同类型的结点所占用的存储单元数量也各不相同。为了简化问题，我们可以通过高级语言的类型说明，使一个结点对应一个存储单元。在以后的叙述中，如果没有特别说明，存储单元都是指高级语言层面上的虚拟存储单元。这样可以用一个一维数组对应一个存储区域，一个数组元素对应一个存储单元。

数据结构包括结点的值及结点之间的关系，其存储结构除了必须存储结点的值外，还应该能以直接或间接的方式体现结点之间的关系。

下面介绍四种基本存储表示方法。

### 1. 顺序存储方式

#### (1) 顺序存储的概念

在顺序存储方式中，逻辑上相邻的结点被存放到物理上相邻的存储单元中。在顺序存储结构中只存储结点的值，不存储结点之间的关系。结点之间的关系是通过存储单元的相邻关系隐含地表示出来的。顺序存储适合于线性结构。

#### (2) 顺序存储的例子

下面举一个线性结构的顺序存储的例子：

$$B = (D, R)$$

$$D = \{A, B, C, D, E, F, G\}$$

$$R = \{ \langle A, B \rangle, \langle B, C \rangle, \langle C, D \rangle, \langle D, E \rangle, \langle E, F \rangle, \langle F, G \rangle \}$$

顺序存储的存储结构如图 1-4 所示。