

● 高等学校21世纪计算机教材

数据结构

(C语言描述)

刘怀亮 编著

冶金工业出版社

高等学校 21 世纪计算机教材

数 据 结 构

(C 语言描述)

刘怀亮 编著

北 京

冶 金 工 业 出 版 社

2004

内 容 简 介

本书系统地介绍了各种常用的数据结构,讨论了它们在计算机上的存储方式和具体的实现方法。全书共9章,第1章绪论部分,介绍了数据结构的一些基本概念和术语,并说明了算法的描述和分析方法;第2章到第5章主要阐述了线性结构中的线性表、栈、队列、串、数组和广义表这几种数据结构;第6章和第7章介绍了非线性结构的树、二叉树和图;第8章和第9章分别讨论了各种查找和排序算法;书末给出了各章综合练习的参考答案。

本书概念清晰、重点突出、详略得当,表达上力求通俗易懂,结构安排上循序渐进。在详尽阐述原理的同时,本书更注重内容的实用性,强调对各种数据结构及其应用的理解。

本书可以作为本科、大专和高职类院校计算机专业的教材,根据内容的取舍也可以作为研究生入学考试、中国计算机专业技术资格和水平考试、计算机等级考试、计算机专业自学考试和各类培训班的教材,同时也可供一切从事与计算机科学与技术有关的科技工作者参考。

图书在版编目(CIP)数据

数据结构: C语言描述 / 刘怀亮编著. —北京: 机械工业出版社, 2004.6
ISBN 7-5024-3539-5

I. 数... II. 刘... III. ①数据结构 ②C语言—程序设计 IV. ①TP311.12 ②TP312

中国版本图书馆 CIP 数据核字 (2004) 第 045913 号

出版人 曹胜利 (北京沙滩嵩祝院北巷 39 号, 邮编 100009)

责任编辑 程志宏

湛江蓝星南华印务公司印刷; 冶金工业出版社发行; 各地新华书店经销

2004 年 7 月第 1 版, 2004 年 7 月第 1 次印刷

787mm × 1092mm 1/16; 18.5 印张; 424 千字; 286 页; 1-5000 册
30.00 元

冶金工业出版社发行部 电话: (010) 64044283 传真: (010) 64027893

冶金书店 地址: 北京东四西大街 46 号 (100711) 电话: (010) 65289081

(本社图书如有印装质量问题, 本社发行部负责退换)

前 言

一、关于本书

“数据结构”是计算机专业一门重要的专业基础课程，同时也是一门核心课程。数据结构不仅为程序设计提供了方法论性质的指导，还在更高层次上总结了现实生活中多种数据的计算机处理方法，因此是一门很重要的软件课程。

本书作者依据计算机专业的相关教学大纲，通过分析国内外多种同类书籍，在教学和工程实践之中编写了这本书，现将之奉献给广大读者。

二、本书结构

全书共 9 章，具体安排如下：

第 1 章：绪论。介绍了数据结构的一些基本概念和术语，并说明了一些算法的描述和分析方法。

第 2 章：线性表。介绍了线性表的定义和基本运算、线性表的顺序存储结构及其运算，线性表的链式存储结构及其运算、一元多项式相加。

第 3 章：栈和队列。介绍了栈和队列的定义、在顺序和链式存储结构上如何实现栈和队列的基本操作、栈与队列的应用。

第 4 章：串。介绍了串的基本概念、串的存储结构和串的基本运算。

第 5 章：数组和广义表。介绍了数组的定义和运算、数组的顺序存储结构及其实现，特殊矩阵和稀疏矩阵的压缩存储、广义表。

第 6 章：树和二叉树。介绍了树的基本概念和术语、二叉树、遍历二叉树、线索二叉树、树和森林、哈夫曼树。

第 7 章：图。介绍了图的基本概念、图的存储结构、图的遍历、最小生成树、最短路径、拓扑排序。

第 8 章：查找。介绍了查找的基本概念、顺序查找、折半查找、分块查找、动态查找表查找、哈希表查找。

第 9 章：排序。介绍了排序的基本概念、插入排序、交换排序、选择排序、归并排序、基数排序和各种内部排序方法的比较。

三、本书特点

本书选材上覆盖了数据结构的各知识要点，并对重点内容进行了详细的阐述。做到内容新颖、概念清晰、讲究实际，表达上力求通俗易懂、循序渐进。

考虑到 C 语言的高效、灵活、精练和它的广泛应用价值，本书描述算法时采用 C 语言。由于 C 语言提供了较强的结构描述功能，本书用 C 语言的基本类型及自定义类型描述数据结构抽象算法，并提供在计算机上实现的方式。

在详尽阐述原理的同时，本书更注重内容的实用性，强调对各种数据结构及其应用的

理解。各章配有大量例题和综合练习，使读者在了解数据结构基本理论的基础上，掌握数据结构的各种算法及其实现和实际的应用。为便于读者自学，书末给出了每一章综合练习的参考答案，使读者可以更深一步地理解数据结构的内容，并能够在机器上练习和实践。

四、适用对象

本书可以作为本科、大专和高职类院校计算机专业的教材，根据内容的取舍也可以作为研究生入学考试、中国计算机软件专业技术资格和水平考试、计算机等级考试、计算机专业自学考试和各类培训班的教材，同时也可供一切从事与计算机科学与技术有关的科技工作者参考。

本书在编写过程中得到了计算机工程研发中心的多位老师、研究生和工程研发人员的热情帮助，在此表示衷心感谢。

由于编写时间仓促，编者水平有限，书中错误和疏漏之处在所难免，恳请广大读者批评指正。

虽然经过严格的审核、精细的编辑，本书在质量上有了一定的保障，但我们的目标是力求尽善尽美，欢迎广大读者和专家对我们的工作提出宝贵建议，联系方法如下：

电子邮件：service@cnbook.net

网址：www.cnbook.net

此外，本书附送的电子教案可从该网站的下载中心免费下载，同时该网站还有一些其他相关书籍的介绍，可以方便读者选购参考。

编 者

2004 年 7 月

目 录

第1章 绪论	1
1.1 数据结构的发展概况	1
1.2 数据结构的基本概念和术语	2
1.3 数据的逻辑结构和存储结构	5
1.3.1 数据的逻辑结构	5
1.3.2 数据的存储结构	6
1.4 算法及其描述形式	8
1.4.1 算法简介	8
1.4.2 算法的描述形式	8
1.5 算法分析	12
1.5.1 算法效率及其度量	12
1.5.2 算法时间复杂度的计算	14
1.5.3 空间复杂度	16
小结	16
综合练习一	16
一、选择题	16
二、填空题	17
三、综合题	18
第2章 线性表	20
2.1 线性表的定义和基本运算	20
2.1.1 线性表的定义	20
2.1.2 线性表的基本运算	21
2.2 线性表的顺序存储结构及其运算	21
2.2.1 顺序表的表示和实现	21
2.2.2 顺序表的基本运算	22
2.3 线性表的链式存储结构及其运算	25
2.3.1 线性单链表及其运算	26
2.3.2 循环链表	36
2.3.3 双链表	38
2.4 一元多项式相加	41
小结	44
综合练习二	44
一、选择题	44
二、填空题	45

三、综合题	46
第 3 章 栈和队列	47
3.1 栈	47
3.1.1 栈的定义	47
3.1.2 栈的顺序存储及其基本运算	49
3.1.3 栈的链式存储及其基本运算	54
3.1.4 栈的应用——算术表达式求值	56
3.1.5 栈与递归	61
3.2 队列	64
3.2.1 队列的定义	64
3.2.2 队列的顺序存储结构——顺序队列	65
3.2.3 循环顺序队列	68
3.2.4 队列的链式存储结构——链队列	71
3.2.5 队列的应用举例	75
小结	80
综合练习三	80
一、选择题	80
二、填空题	81
三、综合题	82
第 4 章 串	84
4.1 串的基本概念	84
4.2 串的存储结构	85
4.2.1 串的顺序存储结构	86
4.2.2 串的链式存储结构	87
4.3 串的基本运算	87
4.3.1 串赋值运算	88
4.3.2 串复制	88
4.3.3 求串长	88
4.3.4 判断串相等	88
4.3.5 串连接	89
4.3.6 求子串	89
4.3.7 子串定位	90
4.3.8 子串插入	90
4.3.9 子串删除	91
4.3.10 子串替换	92
小结	92
综合练习四	93

一、选择题	93
二、填空题	93
三、综合题	94
第 5 章 数组和广义表	95
5.1 数组的定义和运算	95
5.1.1 数组的定义	95
5.1.2 数组的运算	96
5.2 数组的顺序存储结构及其实现	97
5.3 特殊矩阵和稀疏矩阵的压缩存储	99
5.3.1 特殊矩阵及其压缩存储	99
5.3.2 稀疏矩阵及其压缩存储	100
5.4 广义表	110
5.4.1 广义表的定义	110
5.4.2 广义表的存储结构	111
小结	115
综合练习五	116
一、选择题	116
二、填空题	116
三、综合题	117
第 6 章 树和二叉树	118
6.1 树的基本概念和术语	118
6.1.1 树的定义	118
6.1.2 树的常用术语	119
6.1.3 树的表示方法	121
6.1.4 树的运算	121
6.2 二叉树	122
6.2.1 二叉树的基本概念	122
6.2.2 二叉树的基本运算	123
6.2.3 二叉树的基本性质	124
6.2.4 二叉树的存储结构	126
6.3 遍历二叉树	128
6.3.1 遍历二叉树的定义及递归算法	128
6.3.2 遍历二叉树的非递归算法	130
6.3.3 二叉树基本操作举例	133
6.4 线索二叉树	138
6.5 树和森林	142
6.5.1 树的存储结构	142

6.5.2 树和二叉树之间的转换	144
6.5.3 森林和二叉树之间的转换	146
6.5.4 树和森林的遍历	146
6.6 哈夫曼树	148
6.6.1 有关术语及哈夫曼树的定义	148
6.6.2 构造哈夫曼树	149
6.6.3 哈夫曼编码	149
6.6.4 哈夫曼算法的实现	151
小结	153
综合练习六	153
一、选择题	153
二、填空题	154
三、综合题	156
第 7 章 图	159
7.1 图的基本概念	159
7.1.1 图的定义	159
7.1.2 图的有关术语	160
7.1.3 图的基本运算	164
7.2 图的存储结构	164
7.2.1 邻接矩阵	165
7.2.2 邻接表	166
7.3 图的遍历	168
7.3.1 深度优先搜索	168
7.3.2 广度优先搜索	170
7.4 最小生成树	172
7.4.1 Prim (普里姆) 算法	173
7.4.2 Kruskal (克鲁斯卡尔) 算法	175
7.5 最短路径	176
7.5.1 求单源点最短路径的 Dijkstra (迪杰斯特拉) 算法	176
7.5.2 求每对顶点之间的最短路径	179
7.6 拓扑排序	180
7.6.1 拓扑排序的有关概念	180
7.6.2 拓扑排序算法	182
小结	186
综合练习七	186
一、选择题	186
二、填空题	187
三、综合题	190

第 8 章 查找	191
8.1 查找的基本概念	191
8.2 顺序查找	193
8.3 折半查找	194
8.4 分块查找	198
8.5 动态查找表	201
8.5.1 二叉排序树的查找	202
8.5.2 平衡二叉树	210
8.6 哈希表查找	212
8.6.1 哈希表的概念	212
8.6.2 哈希函数的构造方法	214
8.6.3 哈希表处理冲突的方法	217
8.6.4 哈希表查找运算及性能分析	219
小结	224
综合练习八	224
一、选择题	224
二、填空题	225
三、综合题	227
第 9 章 排序	228
9.1 排序的基本概念	228
9.2 插入排序	229
9.2.1 直接插入排序	229
9.2.2 希尔排序	231
9.3 交换排序	233
9.3.1 冒泡排序	233
9.3.2 快速排序	235
9.4 选择排序	238
9.4.1 直接选择排序	238
9.4.2 堆排序	239
9.5 归并排序	245
9.6 基数排序	248
9.7 各种内部排序方法的比较	252
小结	253
综合练习九	253
一、选择题	253
二、填空题	254
三、综合题	255

参考答案	257
第 1 章	257
第 2 章	257
第 3 章	262
第 4 章	267
第 5 章	268
第 6 章	272
第 7 章	275
第 8 章	278
第 9 章	281
参考文献	286

第 1 章 绪 论

内容提要:

本章的主要内容是: 数据结构的基本概念、基本术语、算法的描述和分析。

学习要点:

- (1) 了解数据结构的发展概况。
- (2) 掌握数据结构的基本概念和术语。
- (3) 理解数据的逻辑结构和存储结构的含义。
- (4) 理解数据结构中数据运算的含义。
- (5) 掌握算法的基本概念和重要特性。
- (6) 熟悉用 C 语言描述数据结构的规范和方法。
- (7) 掌握算法的分析方法。
- (8) 掌握估算算法时间复杂度的方法。
- (9) 了解空间复杂度的含义。

1.1 数据结构的发展概况

数据结构 (Data Structure) 是计算机软件和计算机应用等相关专业的一门核心课程, 它所研究的是非数值计算中计算机操作的对象, 以及对象之间的关系和操作等。

要理解数据结构的重要性, 先回顾一下它的发展历程。

计算机发展的初期, 由于其价格昂贵, 运算速度慢, 存储容量有限, 人们使用计算机主要是用于处理数值问题。其特点是计算量大, 数据量小, 程序设计者把主要精力集中在程序设计的技巧上, 无须重视数据结构。

随着计算机技术的发展, “非数值问题” 越来越显得重要。计算机大量应用于非数值计算领域, 如工业领域、商业企业和政府机构的管理等领域。据统计, 当今处理非数值问题占用了 90% 以上的机器时间。这些问题所涉及的数据量大, 数据结构复杂, 数据之间的关系一般无法用数学方程式加以描述。解决此类问题的关键已不再是分析数学和计算方法, 而必须研究数据自身的内部结构, 因而对于数据结构的设计显得尤为重要。

20 世纪 60 年代, 美国计算机界首先使用了数据结构这一名称。1968 年美国著名的计算机专家唐·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》书中的第一卷《基本算法》中全面、系统地阐述了数据的逻辑结构和存储结构, 为数据结构奠定了基础。

从 60 年代末期到 70 年代初期, 软件相对独立, 程序的规模也越来越大。由于软件危机的出现, 结构化程序设计成为程序设计方法学和软件设计的主要内容, 人们越来越重视数据结构。

著名的瑞士科学家沃斯 (N. Wirth) 教授曾提出: “算法+数据结构=程序”, 得到计算机界广泛地认可。这里的数据结构是数据的逻辑结构和存储结构, 而算法则是对数据运算的描述。

人们普遍认为程序设计实质上是对实际问题选择一种好的数据结构, 加上设计一个好的算法, 而好的算法在很大程度上取决于描述实际问题的数据结构。

从 70 年代中期到 80 年代初期, 人们对于数据结构的研究逐渐产生了浓厚的兴趣, 各种版本的数据结构著作相继出现。

数据结构的研究涉及到计算机硬件和软件的许多领域。在硬件方面, 硬件编码理论、存储装置和存储方法等与数据结构有着密切的关系; 在软件方面, 无论是大量的计算机应用软件还是系统软件如操作系统和编译系统等, 都需要很多数据结构的内容, 如数据元素在存储器中的分配等。

目前, 随着许多大型数据库的出现和互联网资源的迅猛发展, 数据大量涌现。如何从大量的数据中检索到自己所需要的东西成为人们必须考虑的东西, 研究信息检索时人们也必须考虑到如何组织数据, 以便使查找和存储数据元素更为方便。

可以认为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核心课程。在计算机科学中, 数据结构不仅是一般程序设计的基础, 而且是设计和实现程序编译系统、计算机操作系统、数据库系统和其他系统程序和大型应用程序的基础, 因此学习数据结构具有十分重要的意义。

1.2 数据结构的基本概念和术语

1. 数据 (Data)

计算机领域所指的数据概念非常广泛, 可以说能输入到计算机并且能够被计算机所处理的一切对象都可以称之为数据。数据是信息的载体, 是对客观事物的符号表示, 它能够被计算机识别、存储、加工和处理, 是对能够有效地输入到计算机中并且能够被计算机处理的符号的总称。例如, 数学中所用到的整数和实数, 文本编辑所用到的字符串等都是数据。

随着计算机软、硬件技术的发展, 计算机能够处理的对象也在扩大, 相应的数据的含义也被拓宽了, 例如, 一些图形、图像、声音等也都属于数据的范畴。

2. 数据元素 (Data Element)

数据元素是数据的基本单位, 有些情况下也称为元素、结点、顶点、记录等。

3. 数据项 (Data Item)

数据项是数据不可分割的最小单元, 是具有独立含义的最小标识单位。例如, 构成一个数据元素的字段、域、属性等都可称之为数据项。

4. 数据结构 (Data Structure)

简而言之, 数据结构是数据之间的相互关系即数据的组织形式。

虽然至今没有一个关于数据结构的标准定义, 但普遍认为数据结构应当包含以下三方面的内容:

1) 逻辑结构 (Logical Structure)

数据的逻辑结构是指数据之间的逻辑关系。逻辑结构是从逻辑关系上描述数据, 与具体存储无关, 是独立于计算机的。因此逻辑结构可以看作是从具体问题抽象出来的数学模型。

2) 存储结构 (Storage Structure)

数据的存储结构是指数据元素及其关系在计算机存储器内的表示。存储结构是逻辑结构在计算机里的实现,或称之为映像。

对机器而言,存储结构是具体的,它依赖于具体的计算机,但一般只在高级语言层次上来讨论存储结构。

3) 数据的运算

数据的运算是指对数据施加的操作,它定义在数据的逻辑结构之上,每种逻辑结构都有一个运算的集合。

常用的数据运算有:查找、排序、插入、删除、更新等操作。

为增加对数据结构概念的理解,下面举例说明上述概念。

例如,设有一个学生情况登记表,如表 1-1 所示。表中每个学生都是以学号为顺序排列的。

表 1-1 学生情况登记表

学号	姓名	性别	年龄	成绩
990701	赵惠惠	女	19	85
990702	张小林	男	21	77
990703	李小波	男	18	82
990704	刘西	女	19	90
990705	张林森	男	21	63
990706	王大山	男	22	76
990707	宋秋波	男	20	59
990708	李春燕	女	19	78
990709	柳传松	男	21	81
990710	刘春霞	女	20	93
990711	张华仔	男	21	78
990712	杨燕妮	女	17	86
990713	刘东冬	男	21	69
990714	丁一杰	男	19	95
990715	周明敏	女	20	68
990716	钟志秋	男	21	58
990717	刘天一	男	22	86

可以把这张学生情况登记表称为一个数据结构,表中的每一行称为一个数据元素(或结点、记录等),它由学号、姓名、性别、年龄和成绩四个数据项组成。该表中数据元素之间的逻辑关系是:表中任一个结点,与它相邻而且在它前面的结点(称为直接前驱, Immediate Predecessor)最多只有一个;与表中任一结点相邻且在其后的结点(称为直接后继, Immediate Successor)也最多只有一个。表中第一个结点称为开始结点,它没有直接前驱。表中最后一个结点称为终端结点,它没有直接后继。例如,表中“王大山”所在结点的直接前驱结点和直接后继结点分别是“张林森”和“宋秋波”所在的结点。上述结点间的关系构成了这张学生情况登记表的逻辑结构。该表的存储结构是指在计算机的存储器中如何表示结点之间的这种关系,即表中的结点是按顺序存储在一片连续的存储单元中的,

还是分别存储在几片不连续的存储单元之中, 然后用指针将它们连接到一起的。

讨论了前面的基本概念之后, 现看一下数据结构 (表 1-1) 的运算。查找操作是数据结构最基本的运算。对于表 1-1, 如果给出学生的学号, 就可以立即得到该学生的基本情况, 因为这张表是按照学号的逻辑顺序排列的。但是如果想知道某分数段的情况, 例如要查找 90 分以上的同学的情况, 这就比较麻烦, 需要从头到尾地进行逐个扫描, 这种操作效率很低, 尤其是当表很大时更为明显。

为了使按照成绩查找操作方便, 可以改变该表的数据结构, 将表 1-1 中所登记的学生情况进行重新组织。

例如, 可以按照优、良、中、及格和不及格的标准, 将成绩在 90 分以上、80~89 分、70~79 分、60~69 分、0~59 分的学生情况分别登记在 5 个独立的子表中, 如表 1-2、表 1-3、表 1-4、表 1-5、表 1-6 所示。

表 1-2 成绩在 90 分以上的学生情况登记表

学号	姓名	性别	年龄	成绩
990704	刘西	女	19	90
990710	刘春霞	女	20	93
990714	丁一杰	男	19	95

表 1-3 成绩在 80~89 分之间的学生情况登记表

学号	姓名	性别	年龄	成绩
990701	赵惠惠	女	19	85
990703	李小波	男	18	82
990709	柳传松	男	21	81
990712	杨燕妮	女	17	86
990717	刘天一	男	22	86

表 1-4 成绩在 70~79 分之间的学生情况登记表

学号	姓名	性别	年龄	成绩
990702	张小林	男	21	77
990706	王大山	男	22	76
990708	李春燕	女	19	78
990711	张华仔	男	21	78

表 1-5 成绩在 60~69 分之间的学生情况登记表

学号	姓名	性别	年龄	成绩
990705	张林森	男	21	63
990713	刘东冬	男	21	69
990715	周明敏	女	20	68

表 1-6 成绩在 0~59 分之间的学生情况登记表

学号	姓名	性别	年龄	成绩
990707	宋秋波	男	20	59
990716	钟志秋	男	21	58

由于改变了表的数据结构, 要查找各分数段的学生情况就比较容易查找, 避免了对全

部学生一遍一遍地扫描。由此可以看出, 可以将数据组织成不同的形式, 从而易于运算, 提高了对数据的处理效率。

综上所述, 可以将数据结构定义为: 按某种逻辑关系组织起来的一批数据, 按一定的方式存储到计算机的存储器中, 并在这些数据之上定义一个运算的集合, 就称之为一个数据结构。

1.3 数据的逻辑结构和存储结构

1.3.1 数据的逻辑结构

对于数据的逻辑结构的描述, 可以采用数学的方法表示。

设 D 表示数据元素 (或称为结点、记录等) 的集合。 R 表示 D 上的关系集合, 即 R 反映了 D 中各元素的前驱和后继关系。

一个数据结构 B 可以表示为如下二元组的形式:

$$B=(D,R)$$

从上式可以看出, 一个数据结构由两部分构成, 表示数据元素的信息 D 和表示各元素之间关系的集合 R 。

在数据结构中, 一般使用二元组的形式表示 D 中各元素之间的前驱和后继关系。设 r 是一个 D 到 D 的关系, $r \in R$, 若 $d, d' \in D$, 且 $(d, d') \in r$, 则 d 是 d' 的前驱, d' 是 d 的后继, 这时候 d 和 d' 是相邻结点 (相对 r 而言); 如果不存在元素 d' 使 $(d, d') \in r$, 则称 d 为终端结点; 如果不存在元素 d' 使 $(d', d) \in r$, 则称 d 为开始结点; 如果 d 既不是终端结点也不是开始结点, 则称 d 为内部结点。

例如, 下面是一年十二个月的数据结构表示方法。

$$B=(D,R)$$

$$D=\{\text{Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec}\}$$

$$R=\{<\text{Jan, Feb}>, <\text{Feb, Mar}>, <\text{Mar, Apr}>, <\text{Apr, May}>, <\text{May, Jun}>, <\text{Jun, Jul}>, <\text{Jul, Aug}>, <\text{Aug, Sep}>, <\text{Sep, Oct}>, <\text{Oct, Nov}>, <\text{Nov, Dec}>\}$$

该数据结构的图形表示法如图 1-1 所示。

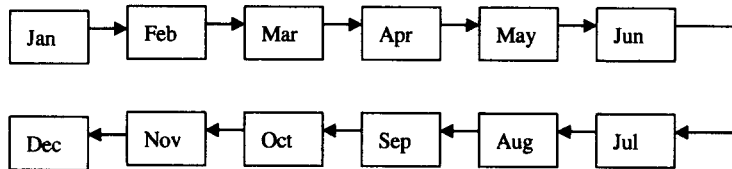


图 1-1 一年十二个月的数据结构

其中, “Jan” 所在的结点 (或称为元素、记录等) 为首结点, “Dec” 所在的结点为尾结点, 其余为内部结点, 结点的前驱和后继关系从图中一目了然。

数据的逻辑结构可以分为以下四种基本的类型:

1. 集合

集合结构中的元素除了仅仅同属于一个集合, 不存在什么关系, 即 $R=\{\}$, 如图 1-2(a) 所示。这里的集合概念和数学上集合的概念是一致的, 由于它是数据结构的一种特例, 本

书不予以讨论。

2. 线性结构

线性结构中的元素是一种一对一的关系, 这种结构的特征是: 若结构是非空集, 则有且只有一个开始结点和一个终端结点, 并且所有结点最多只能有一个直接前驱和一个直接后继, 如图 1-2 (b) 所示。另外图 1-1 和表 1-1 也是一种典型的线性结构, 并且还是一种线性表示特殊的线性结构 (下一章将介绍)。

3. 树型结构

树型结构中的元素是一种一对多的关系。在这种逻辑结构中, 除了根结点外, 其他所有结点有且只有一个前驱, 但是所有结点都可以有多个后继, 如图 1-2 (c) 所示。

4. 图型结构

图型结构或图状结构是一种多对多的关系。在这种逻辑结构中, 所有结点均可以有多个前驱和多个后继, 如图 1-2 (d) 所示。

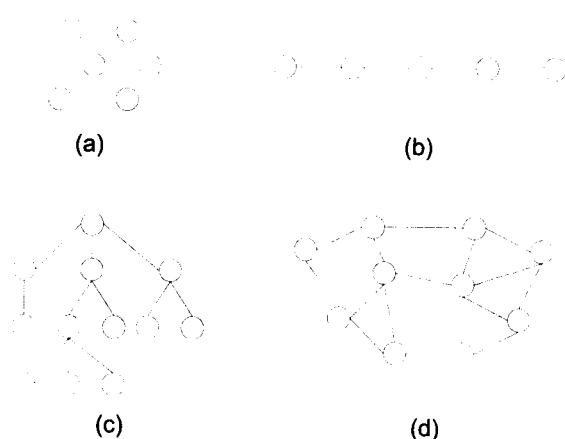


图 1-2 四种基本数据结构图

有时也可将树型结构、集合和图型结构称为非线性结构, 这样数据的逻辑结构就可以分为线性结构和非线性结构两大类。

注意: 在不易混淆的情况下, 常常将数据的逻辑结构简称为数据结构, 后面的章节中将详细讨论它们。

1.3.2 数据的存储结构

数据的存储结构可以用下面四种基本的存储方法得到:

1. 顺序存储方式

顺序存储方式是指逻辑上相邻的结点被存储到物理上也相邻的存储单元中。这种存储结构只存储结点的数值, 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。这种存储方式所得到的存储结构称为顺序存储结构 (Sequential Storage Structure)。

顺序存储结构主要应用于线性的数据结构, 非线性的数据结构也可以通过某种线性化的方法来实现顺序存储。

顺序存储结构通常是借助于程序语言的向量来描述的。在 C 语言中, 通常用数组来表