

数据结构与算法

(C 语言版)

主 编 崔 鹏

副主编 周力波 翟继强

东北林业大学出版社

数据结构与算法

(C 语言版)

主 编 崔 鹏
副主编 周力波 翟继强

东北林业大学出版社

图书在版编目 (CIP) 数据

数据结构与算法: C 语言版/崔鹏主编. —哈尔滨: 东北林业大学出版社, 2006.6
ISBN 7-81076-892-1

I. 数… II. 崔… III. ①数据结构②算法分析③C 语言—程序设计 IV. ①TP
311.12②TP312

中国版本图书馆 CIP 数据核字 (2006) 第 058336 号

责任编辑: 杨秋华

封面设计: 周云凤



NEFUP

数据结构与算法

(C 语言版)

Shuju Jiegou Yu Suanfa

主 编 崔 鹏

副主编 周力波 翟继强

东北林业大学出版社出版发行

(哈尔滨市和兴路 26 号)

哈 尔 滨 理 工 大 学 东 区 印 刷 厂 印 装

开 本 787 × 1092 1/16 印 张 14.25 字 数 320 千 字

2006 年 6 月 第 1 版 2006 年 6 月 第 1 次 印 刷

印数 1—1 000 册

ISBN 7-81076-892-1

TP·69 定价: 21.00 元

内容摘要

本书结合作者教学实践，循序渐进地讲述了数据结构与算法的基本概念原理和应用。全书共分 10 章，主要内容有：数据结构与算法的概述；线性表；栈和队、串；数组和广义表；树和二叉树；图；查找；排序；文件等。本书的特点是在介绍理论的同时，引入了大量具有代表性的例题，通过各种算法实际的应用，提高对理论的理解。

本书既可作为计算机专业或信息类相关专业的本科、专科教学用书，也可供从事计算机工程与应用工作的科技工作者参考。

前 言

随着计算机软件和硬件的发展,计算机的应用已经深入到社会的各个领域,各行各业都需要对大量的非数值数据进行存储、加工和管理。如何根据实际应用的要求,对这些大量的表面上杂乱无章的数据进行有效地组织、存储和处理,编制出相应的高效率算法,这就是“数据结构与算法”这门课程所要研究并加以解决的问题。因此“数据结构与算法”是计算机程序设计的技术基础,它是计算机科学和工程系各专业的核心课程,为计算机专业技术人员提供必要的专业基础知识和技能训练,同时也是计算机应用相关学科所必须掌握的课程。

“数据结构与算法”是计算机专业的专业基础课。通过对该课程的学习,学生能掌握计算机程序设计中常见的各种数据的逻辑结构、存储结构及相应的运算,初步掌握算法的时间分析和空间分析的技术,并能根据计算机加工的数据特性运用数据结构的知识和技巧设计出更好的算法和程序。本课程的学习过程也是复杂程序设计的训练过程,要求学生编写的程序结构清晰明确、通俗易懂,符合软件工程的规范,从而提高学生的软件设计水平。

全书采用类C语言作为数据结构和算法的描述语言。本书文字简练、内容循序渐进、例题丰富,并在各章末配有较多的习题,便于自学。本书所提供的算法不一定是最佳的,仅供读者参考。

本书既可作为计算机专业的本科或专科教材,也可作为信息类相关专业的选修教材,还可作为从事计算机应用相关工作科技人员的参考书。

本书共分10章。第1章介绍数据结构和算法的基本概念,第2、3、4、5章介绍线性结构,第6章介绍树形结构,第7章介绍图形结构,第7章介绍排序方法,第8章介绍查找技术,第9章介绍排序技术,第10章介绍文件的相关概念。其中,前9章为数据结构课程的基本内容,不同的专业可根据需要选择讲解有关内容。

参加本书编写的有崔鹏(第1、2、3、4、5章),周力波(第6、7章),翟继强(第8、9、10章)。东北林业大学机电工程学院的刘晓义参与了本书的构思和制图工作。

由于编写时间仓促,编者水平有限,书中不足之处在所难免,恳请广大读者批评指正。

作 者

2005年10月

目 录

1 绪 论	1
1.1 为什么要学习数据结构	1
1.1.1 数据结构的应用	1
1.1.2 学习数据结构的意义	4
1.2 基本概念和术语	6
1.2.1 数据、数据元素和数据项	6
1.2.2 什么是数据结构	6
1.2.3 数据结构涵盖的内容	7
1.3 抽象数据类型	9
1.3.1 数据类型与抽象数据类型的区别	9
1.3.2 抽象数据类型如何定义	9
1.3.3 抽象数据类型如何表示和实现	10
1.4 算法及其效率的度量	10
1.4.1 什么是算法	10
1.4.2 算法效率	10
习题一	13
2 线性表	16
2.1 线性表的定义和操作	16
2.1.1 线性表的定义	16
2.1.2 线性表的抽象数据类型	18
2.2 线性表的顺序表示和实现	19
2.2.1 顺序表的表示	19
2.2.2 顺序表的实现（或操作）	21
2.2.3 顺序表的运算效率分析	25
2.3 线性表的链式表示和实现	26
2.3.1 链表的表示	26
2.3.2 单链表的实现	31
2.3.3 循环链表的实现	34
2.3.4 双向链表的实现	35
2.3.5 链表的运算效率分析	37
2.4 应用举例	37
习题二	50
3 栈和队列	53
3.1 栈	53
3.1.1 栈的定义	53
3.1.2 栈的基本操作	56
3.1.3 栈的应用举例	58
3.2 队列	60
3.2.1 队列的定义	60
3.2.2 队列的存储结构及其基本操作	60
习题三	66
4 串（STRING）	69
4.1 串类型的定义	69

4.2 串的实现和表示	71
4.3 串的模式匹配算法	73
4.3.1 BF 算法	73
4.3.2 KMP 算法	74
习题四	79
5 数组和广义表	81
5.1 数组的定义	81
5.2 数组的顺序存储表示和实现	82
5.3 矩阵的压缩存储	84
5.3.1 稀疏矩阵的压缩存储	84
5.3.2 稀疏矩阵的操作	86
5.4 广义表的定义	89
5.4.1 广义表的定义	89
5.4.2 广义表的特点	89
5.5 广义表的存储结构	91
习题五	93
6 树和二叉树	96
6.1 树的基本概念	96
6.1.1 树的定义	96
6.1.2 若干术语	96
6.1.3 树的逻辑结构和存储结构	97
6.2 二叉树	98
6.2.1 二叉树的定义	98
6.2.2 二叉树的性质	99
6.2.3 二叉树的存储结构	100
6.3 遍历二叉树和线索二叉树	102
6.3.1 遍历二叉树	102
6.3.2 线索二叉树	105
6.4 树和森林	109
6.4.1 树和森林与二叉树的转换	109
6.4.2 树和森林的存储方式	111
6.4.3 树和森林的遍历	112
6.5 HUFFMAN 树及其应用	117
6.5.1 构造 Huffman 树的基本思想	117
6.5.2 构造 Huffman 树的步骤	117
6.5.3 Huffman 编码	118
习题六	122
7 图	126
7.1 图的基本术语	126
7.2 图的存储结构	130
7.2.1 邻接矩阵(数组)表示法	130
7.2.2 邻接表(链式)表示法	132
7.2.3 十字链表表示法	134
7.2.4 邻接多重表表示法	136
7.3 图的遍历	137
7.3.1 深度优先搜索(DFS)	137
7.3.2 广度优先搜索(BFS)	139
7.4 图的其他运算	142
7.4.1 求图的生成树(或生成森林)	142

7.4.2 求最小生成树	145
7.4.3 求最短路径	146
7.5 有向无环图及其应用	148
7.5.1 拓扑排序	148
7.5.2 关键路径	149
习题七	152
8 查找	158
8.1 基本概念	158
8.2 静态查找表	159
8.2.1 顺序查找	159
8.2.2 折半查找	160
8.2.3 分块查找	161
8.2.4 静态树表的查找	162
8.3 动态查找表	163
8.3.1 二叉排序树的定义	163
8.3.2 二叉排序树的插入与删除	164
8.3.3 二叉排序树的查找分析	166
8.3.4 平衡二叉树	166
8.4 哈希查找表	168
8.4.1 哈希表的概念	168
8.4.2 哈希函数的构造方法	169
8.4.3 冲突处理方法	171
8.4.4 哈希表的查找及分析	173
习题八	176
9 排 序	181
9.1 概述	181
9.2 内部排序的算法	182
9.2.1 插入排序	182
9.2.2 交换排序	186
9.2.3 选择排序	189
9.2.4 归并排序	196
9.2.5 基数排序 (Radix Sort)	196
9.3 外部排序	202
9.3.1 问题的提出	202
9.3.2 外部排序的基本过程	202
习题九	204
10 文 件	208
10.1 有关文件的基本概念	208
10.2 顺序文件	209
10.3 索引文件	209
10.3.1 索引文件的结构特点	209
10.3.2 索引文件的操作特点	210
10.3.3 索引方式	210
10.4 索引顺序文件	211
10.4.1 ISAM 文件	211
10.4.2 VSAM(Vistual Storage Access Method)文件	212
10.5 直接存取文件	213
10.5.1 直接存取文件的特点	213
10.5.2 哈希文件的结构	213

10.5.3 文件的操作	213
10.5.4 优缺点	214
10.6 多关键字文件	214
10.6.1 多关键字文件的特点	214
10.6.2 次索引的组织方法	214
习题十	215
参考文献	217

1 绪 论

计算机科学是一门研究数据表示和数据处理的科学。数据是计算机化的信息，它是计算机可以直接处理的最基本和最重要的对象。无论是进行科学计算或数据处理、过程控制以及对文件的存储和检索及数据库技术等计算机应用领域中，都是对数据进行加工处理的过程。因此，要设计出一个结构好效率高的程序，必须研究数据的特性及数据间的相互关系及其对应的存储表示，并利用这些特性和关系设计出相应的算法和程序。

1.1 为什么要学习数据结构

1.1.1 数据结构的应用

在计算机发展的初期，人们使用计算机的目的主要是处理数值计算问题。当我们使用计算机来解决一个具体问题时，一般需要经过下列几个步骤：首先要从该具体问题抽象出一个适当的数学模型，然后设计或选择一个解此数学模型的算法，最后编出程序进行调试、测试，直至得到最终的解答。例如，求解梁架结构中应力的数学模型的线性方程组，该方程组可以使用迭代算法来求解。

由于当时所涉及的运算对象是简单的整型、实型或布尔类型数据，因此程序设计者的主要精力是集中于程序设计的技巧上，而无须重视数据结构。随着计算机应用领域的扩大和软、硬件的发展，非数值计算问题越来越显得重要。据统计，当今处理非数值计算性问题占用了 90% 以上的机器时间。这类问题涉及的数据结构更为复杂，数据元素之间的相互关系一般无法用数学方程式加以描述。因此，解决这类问题的关键不再是数学分析和计算方法，而是要设计出合适的数据结构，才能有效地解决问题。下面所列举的就是属于这一类的具体问题。

学生信息检索系统。当我们需要查找某个学生的有关情况的时候，或者想查询某个专业或年级的学生的有关情况的时候，只要我们建立了相关的数据结构，按照某种算法编写了相关程序，就可以实现计算机自动检索。由此，可以在学生信息检索系统中建立一张按学号顺序排列的学生信息表和分别按姓名、专业、年级顺序排列的索引表，如图 1-1 所示。

例 1：由图 1-1 构成的文件便是学生信息检索的数学模型，计算机的主要操作便是按照某个特定要求（如给定姓名）对学生信息文件进行查询。

学号	姓名	性别	专 业	年 级
040001	赵宝库	男	计算机科学与技术	2004 级
040002	张淑芬	女	网络工程	2004 级
050301	吴丽娜	女	软件工程	2005 级
050302	张国友	男	网络工程	2005 级
050303	蔡洪国	男	计算机科学与技术	2005 级
030801	何颖	女	计算机科学与技术	2003 级
030802	王辉	男	软件工程	2003 级
030803	李楠	男	系统集成	2003 级
040601	于茵	女	软件工程	2004 级
040602	翟小飞	男	系统集成	2004 级

(a)

蔡洪国	5
何颖	6
李楠	8
王辉	7
吴丽娜	3
于茵	9
翟小飞	10
张国友	4
张淑芬	2
赵宝库	1

(b)

计算机科学与技术	1, 5, 6
软件工程	3, 7, 9
网络工程	2, 4
系统集成	8, 10

(c)

2003 级	6, 7, 8
2004 级	1, 2, 9, 10
2005 级	3, 4, 5

(d)

图 1-1 学生信息查询系统中的数据结构

(a) 学生信息表; (b) 姓名索引表; (c) 专业索引表; (d) 年级索引表

诸如此类的还有电话自动查号系统、考试查分系统、仓库库存管理系统等。在这类文档管理的数学模型中, 计算机处理的对象之间通常存在着的一种简单的线性关系, 这类数学模型可称为线性的数据结构。

例 2: 八皇后问题。在八皇后问题中, 处理过程不是根据某种确定的计算法则, 而是利用试探和回溯的探索技术求解。为了求得合理布局, 在计算机中要存储布局的当前状态。从最初的布局状态开始, 一步步地进行试探, 每试探一步形成一个新的状态, 整个试探过程形成了一棵隐含的状态树。如图 1-2 所示 (为了描述方便, 将八皇后问题简化为四皇后问题)。回溯法求解过程实质上就是一个遍历状态树的过程。在这个问题中所出现的树也是一种数据结构, 它可以应用在许多非数值计算的问题中。

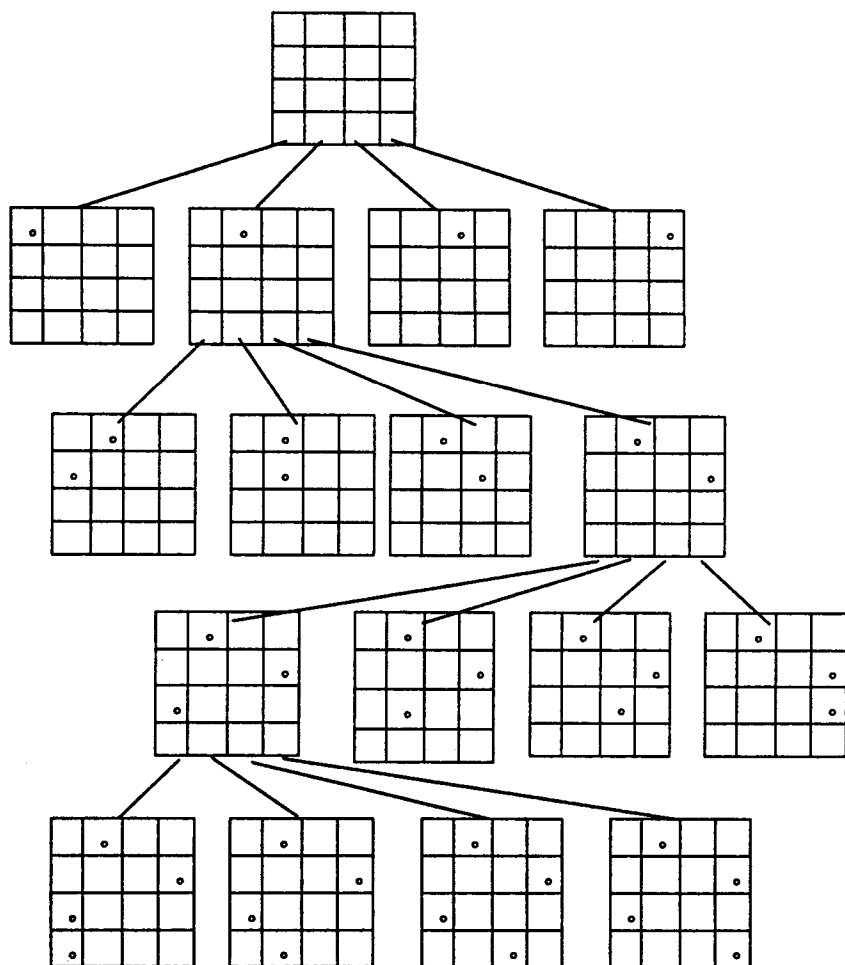


图 1-2 四皇后问题中隐含的状态树

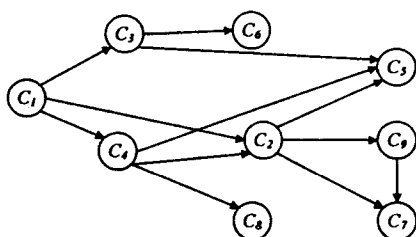
例 3: 教学计划编排问题。一个教学计划包含许多课程, 在教学计划包含的许多课程之间, 有些必须按规定的先后次序进行, 有些则没有次序要求。即有些课程之间有先修和后续的关系, 有些课程可以任意安排次序。这种各个课程之间的次序关系可用一个称做图的数据结构来表示, 如图 1-3 所示。有向图中的每个顶点表示一门课程, 如果从顶点 v_i 到 v_j 之间存在有向边 $\langle v_i, v_j \rangle$, 则表示课程 i 必须先于课程 j 进行。

由以上三个例子可见, 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、图之类的数据结构。因此, 可以说数据结构课程主要是研究非数值计算的程序设计问题中所出现的计算机操作对象以及它们之间的关系和操作的学科。

学习数据结构的目的是为了了解计算机处理对象的特性, 将实际问题中所涉及的处理对象在计算机中表示出来并对它们进行处理。与此同时, 通过算法训练来提高学生的思维能力, 通过程序设计的技能训练来促进学生的综合应用能力和专业素质的提高。

课程编号	课程名称	先修课程
C_1	计算机导论	无
C_2	数据结构	C_1, C_4
C_3	汇编语言	C_1
C_4	C 程序设计语言	C_1
C_5	计算机图形学	C_2, C_3, C_4
C_6	接口技术	C_3
C_7	数据库原理	C_2, C_6
C_8	编译原理	C_4
C_9	操作系统	C_2

(a)



(b)

图 1-3 表示课程之间优先关系的有向图
(a) 计算机专业的课程设置；(b) 表示课程之间优先关系的有向图

1.1.2 学习数据结构的意义

数据结构是计算机科学与技术专业的专业基础课，是十分重要的核心课程。所有的计算机系统软件和应用软件都要用到各种类型的数据结构。因此，要想更好地运用计算机来解决实际问题，仅掌握几种计算机程序设计语言是难以应付众多复杂的课题的。要想有效地使用计算机、充分发挥计算机的性能，还必须学习和掌握好数据结构的有关知识。打好“数据结构”这门课程的扎实基础，对于学习计算机专业的其他课程，如操作系统、编译原理、数据库管理系统、软件工程、人工智能等都是十分有益的。

1.1.2.1 计算机系列课程之间的联系

课前的话——计算机系列课程之间的联系如图 1-4 所示。

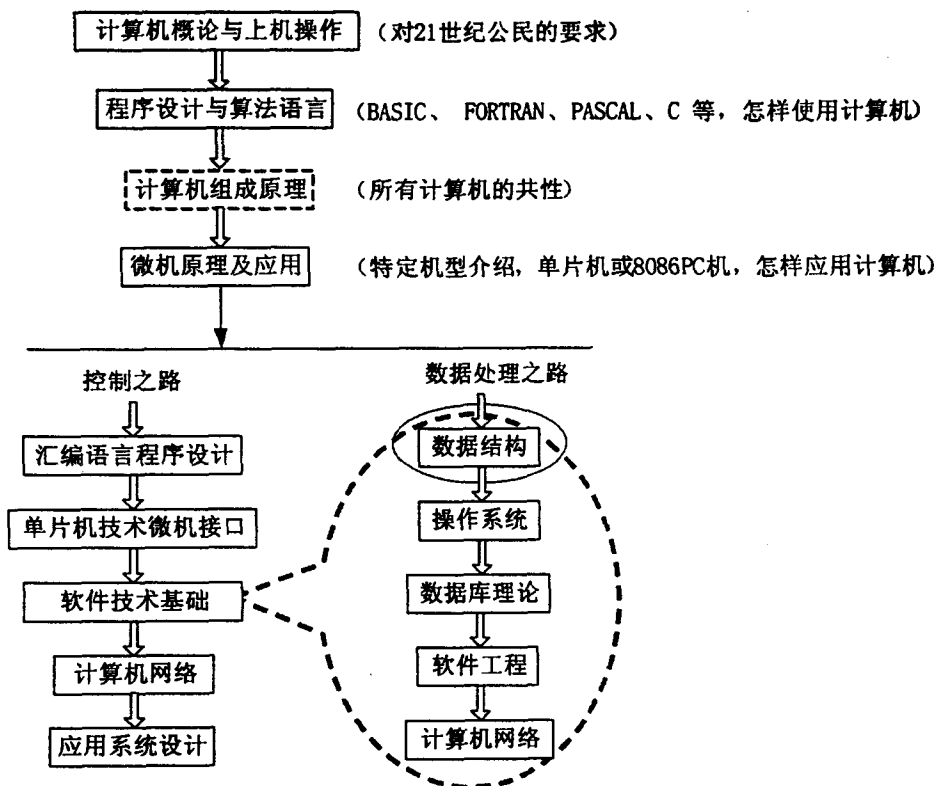


图 1-4 计算机系列课程之间的联系

1.1.2.2 数据结构课程的地位

第一，针对非数值计算的程序设计问题，研究计算机的操作对象以及它们之间的关系和操作。

第二，计算机内的数值运算依靠方程式，而非数值运算（如表、树、图等）则要依靠数据结构。数据结构是一门学科，针对非数值计算的程序设计问题，研究计算机的操作对象以及它们之间的关系和操作等。

第三，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。它们之间的关系如图 1-5 所示：

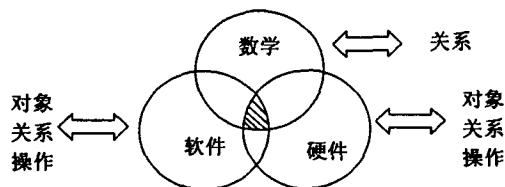


图 1-5 数据结构课程所处的地位

同样的数据对象，用不同的数据结构来表示，运算效率可能有明显的差异。瑞士计算机科学家沃斯(N.Wirth)曾提出“算法+数据结构=程序”。可见，程序设计的实质是对实际问题选择一种好的数据结构，并设计一个好的算法。

1.2 基本概念和术语

在系统地学习数据结构知识之前,先对一些基本概念和术语赋予确切的含义。

1.2.1 数据、数据元素和数据项

数据(Data)——所有能被计算机识别、存储和处理的符号的集合(包括数字、字符、声音、图像等信息)。

数据元素(Data Element)——数据的基本单位,具有完整确定的实际意义(又称元素、结点、顶点、记录等)。

数据项(Data Item)——构成数据元素的项目。是具有独立含义的最小标识单位(又称字段、域、属性等)。

数据、数据元素以及数据项三者之间的关系:数据 > 数据元素 > 数据项。

例:班级通讯录 > 个人记录 > 姓名、年龄

1.2.2 什么是数据结构

数据结构(Data Structure)是指相互之间存在着一种或多种关系的数据元素的集合。在任何问题中,数据元素之间总是存在联系的。把某一数据对象及该数据对象中所有数据成员之间的关系组成的实体叫做数据结构。根据数据元素间关系的不同特性,通常有下列四类基本的结构:

(1) 集合结构。在集合结构中,数据元素间的关系是“属于同一个集合”。集合是元素关系极为松散的一种结构,如蓝色、红色和黄色同属色彩集合。

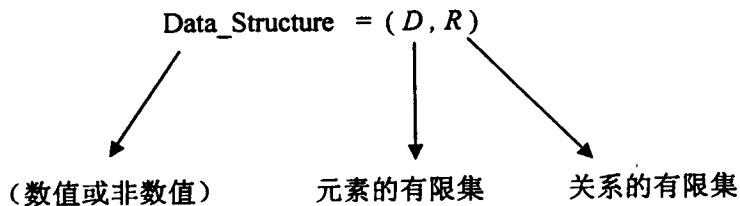
(2) 线性结构。该结构的数据元素之间存在着一对一的关系,如考生信息表中的各元素。

(3) 树结构。该结构的数据元素之间存在着一对多的关系,如家谱,一个父亲对应着多个儿子。

(4) 图结构。该结构的数据元素之间存在着多对多的关系。图结构也称为网状结构,如城市交通图。图 1-6 为表示上述四类基本结构的示意图。

从上面所介绍的数据结构的概念中可以知道,一个数据结构有两个要素。一个是数据元素的集合,另一个是关系的集合。在形式上,数据结构通常可以采用一个二元组来表示。

数据结构的形式定义为:



其中, D 是数据元素的有限集, R 是 D 上关系的有限集。

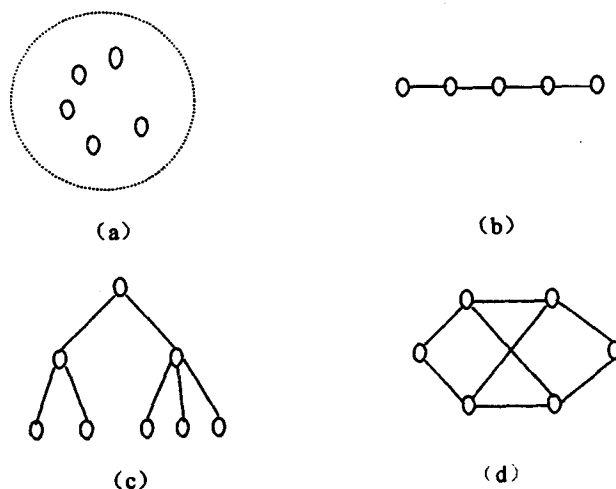


图 1-6 四类基本结构的示意图

(a) 集合结构; (b) 线性结构; (c) 树结构; (d) 图结构

1.2.3 数据结构涵盖的内容

数据结构涵盖的内容如图 1-7 所示。

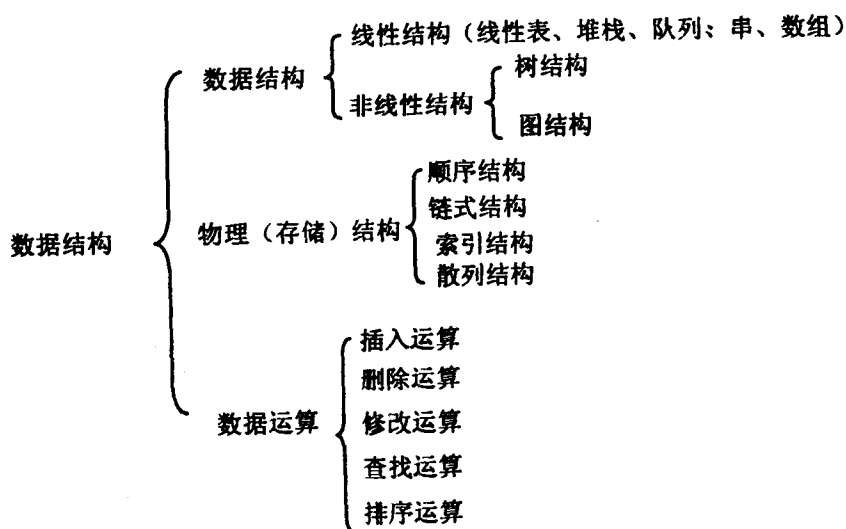


图 1-7 数据结构涵盖的内容

1.2.3.1 数据的逻辑结构

数据的逻辑结构是指数据元素之间的逻辑关系。即从逻辑关系上描述数据，它与数据的存储无关，是独立于计算机的。

如前所述，逻辑结构可细分为四类：

集合结构——仅同属于一个集合；

线性结构——一对一 (1:1)；

树结构——一对多 (1:n)；

图结构——多对多 (m:n)。

树结构和图结构属于非线性结构。

例：用图形表示下列数据结构，并指出它们是属于线性结构还是非线性结构。

(1) $S=(D, R); D=\{a, b, c, d, e, f\}; R=\{(a, e), (b, c), (c, a), (e, f), (f, d)\}$

解：上述表达式可用图形表示为：

$b \rightarrow c \rightarrow a \rightarrow e \rightarrow f \rightarrow d$

每个元素是一一对应关系，所以该结构是线性的。

(2) $S=(D, R); D=\{d_i | 1 \leq i \leq 5\}; R=\{(d_i, d_j), i < j\}$ 。

解：上述表达式如图 1-8 所示。

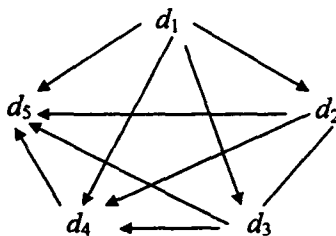


图 1-8 非线性结构

从图 1-8 中可以看出，每个元素并不是一一对应的关系，所以该结构是非线性的。

1.2.3.2 数据的物理结构

物理结构亦称存储结构，是数据的逻辑结构在计算机存储器内的表示（或映像）。它依赖于计算机。

存储结构可分为四大类：顺序、链式、索引、散列。

(1) 顺序存储结构：它是把逻辑上相邻的结点存储在物理位置相邻的存储单元里，结点的逻辑关系由存储单元的邻接关系来体现，由此得到的存储结构称为顺序存储结构。

(2) 链式存储结构：它不要求逻辑上相邻的结点在物理位置上亦相邻。结点之间的逻辑关系是由附加的指针字段表示的。由此得到的存储结构称为链式存储结构。

(3) 索引存储结构：除建立存储结点信息外，还建立附加的索引表来标识结点的地址。

(4) 散列存储结构：根据结点的关键子直接计算出该结点的存储地址。

例：复数 $3.0 - 2.3i$ 的两种存储结构：

结构 1：

地址	内容
0300	3.0
0302	-2.3

} 2 字节

结构 2：

地址	内容
0300	3.0
0302	0415
0415	-2.3

1.2.3.3 数据运算

数据运算是在数据的逻辑结构上定义的操作算法。它在数据的存储结构上实现。最常