

高等学校教材

数据结构与算法导论

徐绪松 编著

TP311.1

出版社

高等学校教材

数据结构与算法导论

徐绪松 编著



电子工业出版社

高等学校教材

数据结构与算法导论

徐绪松 编著

电子工业出版社

内 容 提 要

本书全面系统地阐述了各种数据结构,给出了在各种结构上进行插入、删除、排序、查找的算法及应用实例,介绍了基本的算法分析方法和算法设计技术。全书渗透了作者多年的教学经验、研究心得,写进了作者最新研究成果。

作者将数据结构与算法紧密结合,以“方法——实施方法的数据结构——施加在数据结构上的算法——程序”的模式展开。每章末均配有习题,以起到概括内容、掌握概念、巩固知识的作用。

本书可作为高等学校计算机应用专业、电子工程专业、尤其是管理信息系统、系统工程、情报检索和应用数学、运筹学等专业的教材和参考书。也可供从事上述专业的工作人员自学参考。

高等学校教材

数据结构与算法导论

徐绪松 编著

责任编辑:陈礼璐(特约) 赵家鹏

*

电子工业出版社出版

北京市海淀区万寿路 173 信箱 (100036)

电子工业出版社发行 各地新华书店经销

北京大中印刷厂印刷

*

开本: 787×1092 毫米 1/16 印张: 23.75 字数: 589.7 千字

1996 年 12 月第一版 1996 年 12 月北京第一次印刷

印数: 0, 001—5 000 册 定价: 30.00 元

ISBN 7-5053-3637-1/TP·1501

序

在诸多的同类书籍中，本书具有自己的独特之处。为此，本人谨向读者推荐这本书。概括起来，本书的特点可以列举如下：

1 在整体结构上，作者依据数据类型的性质及相关性，将全书分为五篇、共16章，对数据进行了全局性组合，章节关联体现强的逻辑体系，层次清晰，结构严谨。有利于数据结构及其算法的结合这一概念的逐步深化和完善。

2 对每类数据按逻辑定义、操作定义、物理组织、算法分析和过程实现的模式展开，并以实例说明，贯穿始终。概念明确，解说精辟，图文并茂，易于理解和接受。

3 全书最显著的特点是：作者着力于数据结构及其算法实现的结合，精心于算法的设计，并以自然语言进行描述。对每类数据及其操作的物理组织的动态实现过程，作者从算法的展开进程，以图示法展示于书面上，直观生动，溶宏观描述和微观过程于一体。

4 全书搜集了大量经济信息管理的实例，有很强的实用性。作者对这些实例作了系统的布局，现用来作为说明数据结构的属性，同时也可以看成每个实例都是一个客观对象，而对应的数据结构及其操作是这些对象的具体体现。

5 在排序与查找的内容中，引入了作者有创见性的研究成果，正式开辟了公式分组算法的章节，描述了该算法效率计算公式和结论。此外，在这些章节中，作者还引入了微型计算机磁盘操作系统当代最新的概念和成果。本书在内容的选材和编排上有其先进性。

归结起来，以上这些特点，使该书立于一个新的高度，具有实用性、先进性和科学性。以此书作为教材，将获得教学良好的效果，讲授生动有趣，学生易于理解和接受，并在潜移默化中，使学生掌握数据类型开发、设计、实现和操作的能力，达到学以致用之目的。

李传湘 1995年7月

前 言

随着计算机的发展,计算机已广泛应用于各个领域。近年来兴起的**管理信息系统**、**智能化的决策支持系统**,又开创了计算机应用的新领域。它们不再拘泥于科学计算,而更多的是**数据处理**和**大型离散问题**的程序设计。这种非数值问题,不是一个简单的编程,它将涉及到**数据之间的结构关系**、**计算机内的存储表示**,及在各种结构上的非数值运算(如插入、删除、排序、查找等)。非数值问题占用了90%以上的计算机时间,这就需要研究**数据结构**及其相应的**算法**。

数据结构与算法之间有着密切的联系,不了解施于数据上的算法就无法决定数据结构;反过来,算法的结构和选择在很大程度上依赖于作为基础的数据结构。可以说,数据结构为算法提供了工具,而算法则是运用这些工具来实施解决问题的最优方案。在1987年作者将数据结构与算法有机结合,编著出版了《数据结构与算法》一书(以下简称《数》书)。之后一直致力于数据结构与算法的研究,得到了许多新的启示,尤其是体会到数据结构与算法无论是对计算机科学本身,还是对以计算机为基础的信息系统都起着十分重要的作用。如今站在信息系统这一高度,在已出版的《数》书的基础上,重新编著了《数据结构与算法导论》,再次奉献给读者,渴望得到读者以及同行专家的品评。

作为一本新书,是在计算机广泛应用于经济管理、系统工程、情报检索等领域的新形势下编著的,因此它具有新的特点:①在指导思想,是站在信息系统的高度;②在体系结构上,赋予了数据结构与算法新的体系;③在学术研究上,建立了数据结构与算法相结合的研究模式;④在内容上,写进了作者最新的学术研究、项目研究的研究成果,以及知识创造者的智力活动。⑤在编排上,遵循教学规律,溶入教学经验,图文并茂,丰富实例,力求便于自学。

全书分为16章,由五个部分组成。第一篇 概论(第一章~第三章),阐述了数据结构、算法的基本概念,提出了计算机科学是数据结构与算法的学问,建立了数据结构与算法结合研究非数值问题的研究模式。第二篇 几种逻辑结构(第四章~第九章),介绍了几种逻辑结构——线性表、栈和队、串、数组、树、图,及计算机存储方式、运算和典型应用实例。第三篇 数据运算(第十章~第十二章),讨论了排序、查找、集合操作等数据处理中的重要问题,给出了算法及其复杂性分析。第四篇 文件(第十三章~第十四章),介绍了各种文件组织方式、文件排序、文件应用实例。第五篇 算法设计与分析(第十五章~第十六章),提出了算法复杂性分析的基本方法,算法设计的六个基本策略及在经济管理中的应用实例。除第一篇外,其余各章均附有丰富的习题供读者选用。最后附录给出了四个应用实例的C语言参考程序。

本书在算法描述上选择了自然语言,这样既使得更多的读者易于接受,又可以根据需要很容易翻译成高级语言程序上机运行。

程序设计是从事计算机科学及其应用领域工作的基本功。作者将数据结构与算法结合,以“方法→实施方法的数据结构→施加在数据结构上的算法→程序”的模式展开,用数据

结构与算法的观念指导编程,不仅程序结构性强,易理解,而且对读者编程能力的培养将起到激发引入的效果。

全书内容丰富,而在叙述上深入浅出。对每个算法从问题入手,用图解法模拟计算机的存储及其操作,从而分析操作的规律性,写出通式,最后导出算法,使程序设计水到渠成,对培养学生的分析思维能力大有裨益。尤其结合经济管理的应用实例,不仅激发读者的学习兴趣,更加深对算法的理解,且使之便于自学。

本书站在数据结构与算法的前沿,写进了最新研究成果,它们是作者近年来在排序、查找和最小生成树等非数值问题方面所取得的一些可喜成果。如:和周建钦同志合作研究的公式分组排序、公式分组查找等算法,分别将当前世界上排序的计算复杂度由 $O(n\log n)$ 降到 $O(n)$ 、查找的计算复杂度由 $O(n\log n)$ 降到 $O(n)$ 。同时最小生成树算法、工序问题的动态规划算法等,均将当前同类问题的计算复杂度降低了一个数量级。读者不仅可以从中学习科学的研究方法,并可将它应用于计算机科学本身及现代化管理中。

凡学过一门高级语言程序设计的人都能看懂此书,学到许许多多程序设计技巧。本书还有一部分内容具有一定的深度(*号处的章节),仅供阅读、参考,而不作教学要求。另外,作者认为应十分注重算法分析,因为不仅要培养读者的编程能力,还要给读者树立时间和空间的观点,设计高效的优化算法。由于这是一种创造性的思维活动,有些算法分析具有相当的深度和难度,故这些部分也仅供研究人员参考,而不作教学要求。

书中的所有算法及应用实例都已编写成C语言程序上机调试通过,这一工作是刘大成做的。第十章的公式分组排序和第十一章的公式分组查找的实验结果是刘大成所做。附录中的应用程序一由汪年俊调试,应用程序二、三、由吕晶调试,应用程序四由刘大成调试。郭峰、郑颂阳帮助整理了排序、查找、文件组织等章节的底稿。彭运芳、张襄武帮助整理了1~7章的底图。他们均是武汉大学管理信息系统方向研究生。此外,郑湛帮助整理了其余各章节的底图。

华中理工大学崔国华副教授审阅了全部初稿,北京航空航天大学张关康教授进行了终审,提出了许多宝贵意见。此外,本书的出版得到曾宪昌、李传湘、李光琳、龙其豪等教授及王子丁副教授,张肃文、胡心如、甘良才、胡则成等教授以及武汉大学教务处陶梅生处长给予的真诚帮助、关怀和指导。在此一并表示衷心感谢。

为了读者学习的方便,另配有一本与本书对照的教材《数据结构与算法习题解析》,也一并由电子工业出版社出版发行。

尽管本书耗费了作者10余年的心血,然而错误、不足仍在所难免,敬请专家、读者,批评、指正。

徐绪松 1995年4月
于武汉大学

目 录

第一篇 概 论

第一章 数据结构	(3)
1.1 什么是数据结构	(3)
1.2 数据的逻辑结构	(4)
1.3 数据的物理结构	(5)
第二章 算法	(7)
2.1 什么是算法	(7)
2.2 算法的评价	(7)
2.3 算法的描述	(9)
第三章 数据结构与算法	(11)
3.1 数据结构与算法.....	(11)
3.2 算法+数据结构=程序.....	(11)

第二篇 几种逻辑结构

第四章 线性表	(13)
4.1 线性表.....	(13)
4.2 线性表的顺序存储结构及其运算.....	(14)
4.2.1 向量	(14)
4.2.2 对向量的插入和删除运算.....	(14)
4.2.3 线性表的应用实例——仓库管理系统(一)	(16)
4.3 线性表的链式存储结构及其运算.....	(18)
4.3.1 单链表	(19)
4.3.2 对单链表的插入和删除运算	(20)
4.3.3 线性表的应用实例——仓库管理系统(二)	(24)
4.3.4 循环链表	(27)
4.3.5 多项式的算术运算	(28)
4.3.6 双向链表及其对它的插入和删除	(32)
4.3.7 双向链表与动态存储管理.....	(34)
4.3.8 指数存储管理系统	(40)
4.3.9 双向链表的应用实例——自动预订飞机票系统	(44)
第五章 栈和队列	(51)
5.1 栈.....	(51)
5.2 栈的应用实例.....	(53)
5.2.1 学生业务档案系统	(53)
5.2.2 计算表达式	(54)

5.3	队	(57)
* 5.4	队的应用实例——排队模拟	(60)
第六章	串	(67)
6.1	串的基本概念	(67)
6.2	串的运算及其实现	(67)
6.3	串的存储结构	(70)
* 6.4	模式匹配	(74)
第七章	数组和广义表	(80)
7.1	数组及其存储结构	(80)
7.2	稀疏矩阵的存储压缩及其运算	(81)
7.2.1	求转置矩阵	(82)
* 7.2.2	求矩阵的乘积	(85)
7.3	稀疏矩阵的十字链表结构	(88)
7.4	广义表 and 多重链表	(91)
第八章	树	(94)
8.1	基本术语	(94)
8.2	树的存储结构	(95)
8.3	二叉树	(95)
8.3.1	二叉树的定义	(96)
8.3.2	二叉树的基本性质	(96)
8.3.3	二叉树的存储结构	(98)
8.4	递归与二叉树遍历	(99)
8.4.1	递归	(99)
8.4.2	先序遍历	(100)
8.4.3	中序遍历	(102)
8.4.4	后序遍历	(104)
8.5	线索树	(106)
8.6	树的二叉树表示和运算	(109)
8.6.1	树的二叉树表示	(109)
8.6.2	树的插入和删除	(110)
8.7	树的应用	(114)
8.7.1	二叉排序树	(114)
* 8.7.2	最优叶子搜索树	(116)
8.7.3	判定树	(120)
* 8.8	二叉树的应用实例——银行财务实时处理系统	(122)
第九章	图	(129)
9.1	基本术语	(129)
9.2	图的存储结构	(130)
9.2.1	邻接矩阵	(130)

9.2.2	邻接表	(131)
9.2.3	邻接多重表	(132)
9.3	图的遍历和求图的连通分量	(133)
9.3.1	深度优先搜索	(134)
9.3.2	宽度优先搜索	(136)
9.3.3	求图的连通分量	(137)
9.4	生成树和最小花费生成树	(137)
9.5	最短路径	(141)
9.5.1	从某个源点到其余各顶点的最短路径	(141)
9.5.2	每一对顶点之间的最短路径	(144)
9.6	AOV-网与拓扑排序	(145)
9.7	AOE-网与关键路径	(149)
* 9.8	关键路径的应用实例——制定最优的计划方案	(152)

第三篇 数据运算

第十章	排序	(162)
10.1	插入排序	(162)
10.2	选择排序	(164)
10.2.1	简单选择排序	(164)
10.2.2	堆积排序	(165)
10.3	交换排序	(169)
10.3.1	冒泡排序	(169)
10.3.2	快速排序	(171)
10.4	归并排序	(174)
2路归并排序		
10.5	分配排序	(177)
基数排序		
* 10.6	公式分组排序	(178)
10.6.1	公式分组索引排序	(179)
10.6.2	一次到位排序	(183)
10.6.3	分布排序	(188)
* 10.7	字典排序	(190)
10.7.1	不等长字符串序列的字典排序	(190)
10.7.2	不等长字符串序列的公式分组字典排序	(197)
第十一章	查找	(203)
11.1	基于比较的查找	(203)
11.1.1	顺序查找	(203)
11.1.2	二分查找	(204)
11.1.3	分块查找	(205)
* 11.2	公式分组查找	(206)

11.2.1	公式分组索引查找	(206)
11.2.2	分布查找	(211)
第十二章	集合操作	(214)
12.1	对集合的基本操作	(214)
12.2	顺序搜索和链表结构	(214)
12.3	二元搜索与二元搜索树	(215)
* 12.4	最佳二元搜索树	(216)
* 12.5	UNION-FIND 操作	(220)
12.6	字典和优先队	(225)
12.7	Hash (杂凑) 技术	(229)
12.7.1	Hash 函数的构造方法	(231)
12.7.2	冲突的处理	(233)
* 12.8	集合运算的应用——一个新的最小生成树算法	(236)
第四篇 文件		
第十三章	文件组织	(241)
13.1	基本知识	(241)
13.2	顺序文件	(245)
13.2.1	顺序文件的查找	(245)
13.2.2	顺序文件的更新	(245)
13.3	随机处理文件	(245)
13.3.1	索引文件	(245)
13.3.2	索引顺序文件	(246)
13.4	直接存取文件	(249)
13.4.1	按桶散列	(249)
* 13.4.2	可扩充散列	(250)
13.5	索引链接文件	(251)
* 13.6	多关键字文件	(252)
13.6.1	多重链表文件	(253)
13.6.2	倒排文件	(254)
* 13.7	文件的应用实例——旅馆管理系统	(254)
*第十四章	文件排序	(262)
14.1	文件排序的主要过程	(262)
14.2	磁盘排序	(263)
14.2.1	k 路归并	(263)
14.2.2	并行操作的缓冲区处理	(265)
14.2.3	初始归并段的产生	(273)
14.3	磁带排序	(275)
14.3.1	平衡归并排序	(278)
14.3.2	多步归并排序	(279)

第五篇 算法设计与分析

第十五章 算法分析技术	(281)
15.1 算法分析简介	(281)
15.2 循环程序的分析	(282)
15.3 递归算法的分析	(283)
15.3.1 递归方程	(283)
15.3.2 递归算法的分析	(285)
第十六章 算法设计的技术	(289)
16.1 分割求解法	(289)
求集合的最大、最小元	
16.2 动态规划	(292)
16.2.1 单源路径问题	(292)
16.2.2 资源分配问题	(294)
16.3 子目标法	(298)
吉普车问题	
16.4 探索法	(300)
旅游花费问题	
16.5 回溯法	(301)
组合锁问题	
16.6 分枝与限界	(304)
旅游路线问题	
附 录	(315)

第一篇 概论

电子计算机已广泛地应用于系统工程、运筹学、经济管理……等各个领域，尤其是近年来兴起的一个新的学科——管理信息系统，它们将数据处理向前推进了一步。看几个例子。

例 1 高校教师的教学管理

高校教务处使用计算机对全校教师的教学情况作统一管理。它要了解各门课程由哪些教师承担、每门课程的教学时数、使用教材及各位教师承担的课程、教师的基本情况、教学对象、教学效果等。于是，将此问题抽象成一张表。如表 1 所示。

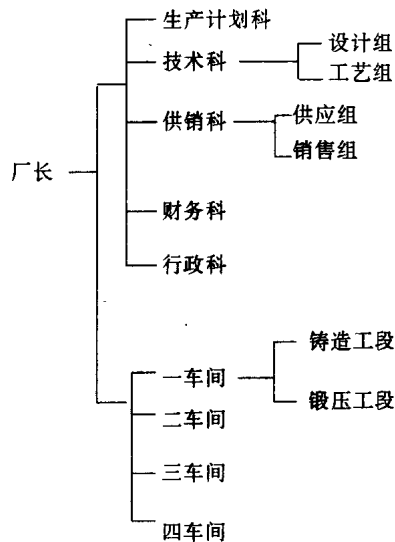
表 1 高校教师教学情况表

姓名	性别	出生年月	职务	工作单位	专业	学历	教龄	所授课程	学时	教学对象	教材	效果
----	----	------	----	------	----	----	----	------	----	------	----	----

每个教师的教学情况按表存入计算机内，根据需要对这张表进行查找。求解这个问题的数学模型是信息表，运算是和信息表的查找操作。

例 2 工厂的组织管理

某工厂的组织机构如下所示。



厂长要通过计算机了解各个科组工作情况及车间生产情况。于是，这个问题可以抽象成如图 1 所示的一棵树。

科室及车间情况按图 1 的树以一定的方式存入计算机内，对这棵树进行遍历便能了解厂内的整个情况。本问题的数学模型是树，运算是树的遍历。

例3 最短路径问题

从油田铺设管道，把原油运到加工厂。求使管道总长最短的铺设方案。这个问题抽象成如图2所示的有向网路。其中 v_1 为油田， v_9 为原油加工厂， $v_2 \sim v_8$ 是问题要求管道必须按给定的道路铺设所经过的地点。每条边旁的数字是这条道路的长度。用计算机求解最短长度的铺设方案，首先要把图2的有向网络按一定方式存入计算机内，然后对这个有向网络设计一种算法（并非简单的数值计算），在各种铺设方案中选一种总的长度最短的铺设方案。那么该问题的数学模型是图，它的运算是图施行一种较为复杂的非数值计算。

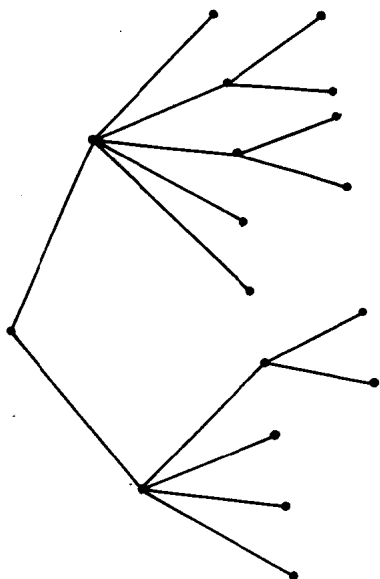


图1 某工厂的组织机构图

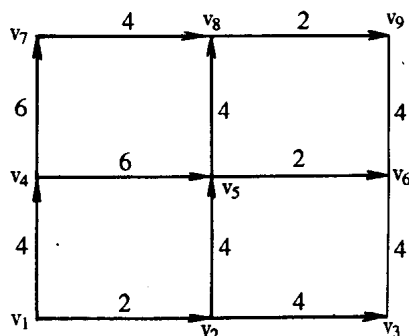


图2 从油田到原油加工厂的有向图

从以上三个例子可以看到，首先描述这类问题的数学模型不再只是数值方程，而是诸如表、树和图的非数值性的数据结构；其次，求解这类问题不再只是数值计算，而是要对一些信息表进行插入、删除、排序、查找；对树进行遍历；对图作较为复杂的非数值计算等。总的来说，当前，计算机面临大量的非数值性程序设计问题。当计算机面临这些非数值性问题时，操作的对象——数据，将具有一定的结构关系，有些甚至具有很复杂的结构关系。因而，对非数值性程序设计需解决如下问题：数据间的结构关系如何表示；数据在计算机内如何存储；处理这些数据（或叫数据运算）有哪些技巧。这就是本书要研究的问题。

这一篇我们将介绍有关数据结构、算法的基本概念，阐述数据结构与算法之间的本质联系，建立一个非数值问题程序设计的模式。本篇的内容将在以后各章节逐步深化、展开。

第一章 数据结构

1.1 什么是数据结构

什么是数据结构？

首先了解一下什么是数据。直观地说，数据是描述客观事物的数字、字母和符号，是计算机程序使用和加工的“原料”。数据的基本单位是数据元素，性质相同的数据元素的集合叫做数据对象。数据对象中的元素彼此之间存在的相互关系叫做结构。

为了进一步理解什么是数据结构，我们来看一个具体例子。图书馆是大家所熟悉的，那么图书馆的结构是什么呢？从物理上来看，图书馆主要是由装书的书架及书籍所组成，但另一方面，还存在一个图书馆的编目表，即图书馆藏书的索引。图书馆从两个方面管理图书：物理的藏书和逻辑的编目表。这就是图书馆的结构。和图书馆一样计算机管理数据，也有两个方面：即物理的存储和逻辑的关系。从这两个方面，我们来回答什么是数据结构。

数据结构指的是数据之间的结构关系。具体来说，它包括数据的逻辑结构和数据的物理结构。

数据的逻辑结构——仅考虑数据元素之间的逻辑关系。它包括**线性结构**（如线性表、栈、队）和**非线性结构**（如树、二叉树和图）。

数据的物理结构——指数据元素在计算机存储器中的表示，即存储结构。比如向量、链表。

一种逻辑结构通过映象便得到它相应的存储结构。同一种逻辑结构可以映象成不同的内部存储结构。反过来，数据的存储结构一定要反映数据之间的逻辑关系。

为了更具体一些，举一个例子。

有一叠扑克牌，希望在计算机中表示这一叠扑克牌的内容（也就是这一组信息）。

在计算机内一组信息是由一组结点组成。在这里我们用一个结点表示一张牌，为了说明这张扑克牌的内容，必须将它的花色（梅花、方块、红心、黑桃）、点数、正反面、名称表示出来。同时还要将这张牌的下一张牌表示出来。为此，用五个域组成一个结点。如图 1.1 所示。

tag	suit	rank	next
title			

图 1.1 结点的格式

其中 tag 表示牌的正、反面（用 0、1 表示），suit 表示花色（用 1、2、3、4、分别表示梅花、方块、红心、黑桃），rank 表示点数，next 表示下一结点的地址（即组成该结点的那些存储单元的首地址），title 表示这张牌的名称，用五个字符表示。

如图 1.2 所示的一叠扑克牌，它的逻辑结构是线性表：（方块 2，梅花 3，黑桃 10（反）），在计算机中相应的存储结构用链表表示，如图 1.3 所示（也可以用别的存储方式）。

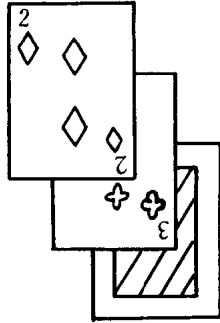


图 1.2 扑克牌的逻辑结构

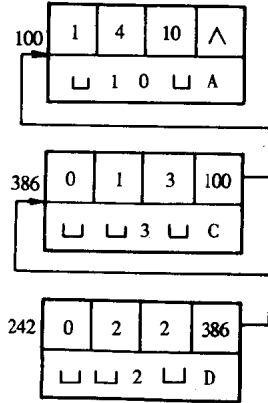


图 1.3 扑克牌的存储结构

1.2 数据的逻辑结构

数据的逻辑结构是独立于计算机的，对数据元素之间的逻辑关系的描述。从集合的观点，它可以形式地用一个二元组 $B = (D, R)$ 表示，其中 D 是数据元素的集合， R 是 D 上关系的集合。

按集合的观点，数据的逻辑结构有两个要素：一是数据元素；二是关系。

数据元素是独立的信息。它可以是一个单独的符号，如英文字母表 (A, B, \dots, Z) ，数据元素是一个字母；也可以由若干个数据项组成，如一个企业单位的全体职工档案登记表（表 1.1），数据元素由姓名、职工号、性别、职务、工资五个数据项组成。

表 1.1 职工档案登记表

姓 名	职 工 号	性 别	职 务	工 资
陈 琳	820721	男	厂 长	1040
王 玉 英	820722	女	工 程 师	1360
刘 萍	820723	女	会 计 师	850
张 健	820724	男	工 人	770
⋮	⋮	⋮	⋮	⋮

关系是指数据元素间的逻辑关系。可以是线性的——对数据元素而言，只有一个前趋和一个后继；也可以是非线性的——对数据元素而言，或者有一个前趋，多个后继；或者有多个前趋，多个后继。

数据的逻辑结构按关系分为线性结构（关系是线性的）和非线性结构（关系是非线性

的)。线性结构包括**线性表**（典型的线性结构。如表 1.1 的职工档案登记表是一个线性表的例子）、**栈和队**（特殊的线性表。是具有特殊限制的线性结构，特殊限制是指数据运算只能在表的一端或两端进行）、**串**（也是特殊的线性表，其特殊性表现在它的数据元素仅由一个字符组成）、**数组**（是线性表的推广，它的数据元素是一个线性表）、**广义表**（也是线性表的推广，它的数据元素是一个线性表，但不同构，即或者是单元素，或者是列表）。非线性结构包括**树**（具有多个分枝的层次结构）和**二叉树**（具有两个分枝的层次结构）、**有向图**（一种网状结构，边是顶点的有序对）和**无向图**（一种网状结构，边是顶点的无序对）。

几种逻辑结构我们用一个层次图描述，如图 1.4 所示。

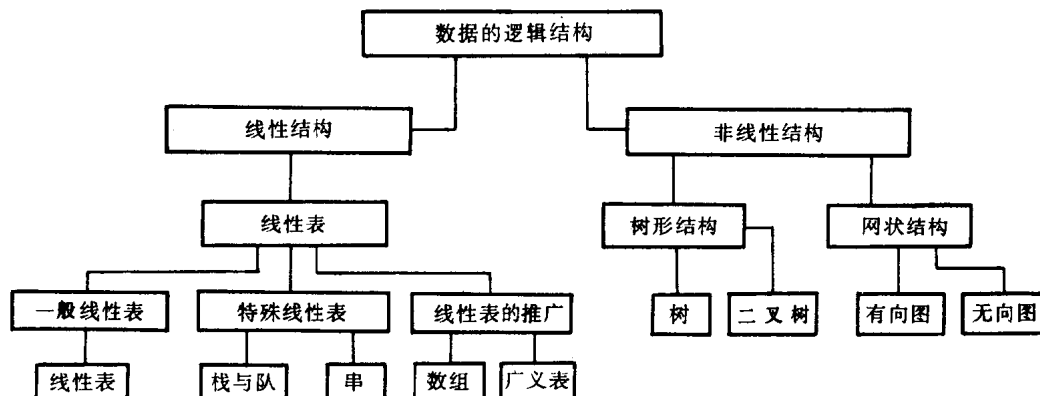


图 1.4 几种逻辑结构

几种逻辑结构（层次图的最底层）的具体定义，在计算机中的存储方式以及数据运算，应用实例等将在第二篇作详细介绍。

1.3 数据的物理结构

数据的物理结构是指数据的逻辑结构在计算机中的映象，即存储表示。映象包括**数据元素的映象**和**关系的映象**。数据元素的映象是**结点**，即在计算机内用一个结点表示一个数据元素（结点是数据结构讨论的基本单位）。关系的映象有两种，**顺序映象**和**非顺序映象**。

数据的物理结构，即存储结构，按关系的映象分为**顺序存储结构**和**非顺序存储结构**。顺序存储结构是逻辑上相邻的数据元素存储在物理位置上相毗邻的存储单元里，元素的关系由存储单元的邻接关系来体现。非顺序存储结构是数据元素可以在计算机内任意位置上存放（它不要求逻辑上相邻的元素在物理位置上也相邻），它们的逻辑关系用**指针**来链接。所以非顺序存储结构又叫**链式存储结构**。链式存储结构将数据元素存放的存储单元分为两个部分，分别用来存放数据和指针，称为**数据域**和**指针域**。

顺序存储结构包括**向量**（典型的顺序存储结构，一组相互毗邻的连续单元）、**数组**。链式存储结构是**链表**，按指针域的个数分为单链表，双向链表和多重链表。

在链式存储结构中，我们提到一个概念——**指针**，指针即地址。程序中的变量名、下标地址都是指针。指针是数据结构中十分关键的概念，对它的理解及其应用都非常重要。首先指针是许多数据结构得以实现的基础，链式存储结构就是用指针来实现逻辑结构与存储结构的映象；其次指针的应用，将导致许多优雅算法，例如，应用指针数组（索引），作

为中间媒介，可不去移动真实的数据，从而利用好计算机程序的时间和空间两大资源。

无论是什么样的存储结构，实际上计算机系统仅为我们提供了一个最基本的存储方式，即一维数组。也就是任何计算机系统的主存可以看作是一个一维数组。尽管大多数高级语言的编译程序还向我们提供了二维数组、三维数组，但实际存储仍是一组连续单元。比如，程序员定义了一个二维数组 A (4, 7)，对用户来讲他可以把内存看作是 4×7 的二维数组结构，但在内存实际上是作为 28 个连续的存储单元存在的。那末程序员的这个逻辑观点和内存的实际分配之间是如何联系的呢？这种联系是通过程序设计语言来完成的。即通过一个下标计算公式将二维数组的下标 (i, j) 映成一维数组的下标。比如，在 FORTRAN 中，一个二维数组按列分配，见图 1.5。在程序访问二维数组 A (i, j) 时，利用下标公式 $4 \times (j-1) + i$ 的计算结果进行访问。

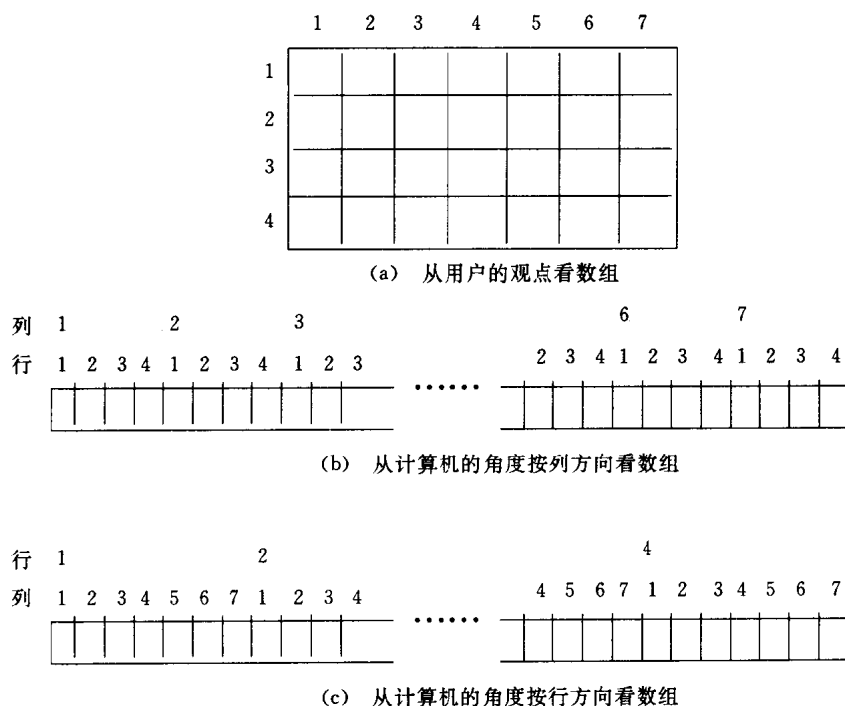


图 1.5 二维数组存储的逻辑图和物理图

较高维数的数组，同样按一维数组存储，通过建立的各种下标计算公式进行访问。

所以，任何一种存储结构都有两种状态，一种是逻辑状态（用户的观点，如图 1.5 (a) 所示），一种是物理状态（计算机的角度，如图 1.5 (b)、(c) 所示）。

最后，值得指出的是，选择某个结构和选择某个结构的表示是不同的。前者是为了解决某个问题，在对问题理解的基础上，选择一个合适的逻辑结构表示出数据的逻辑关系；后者是对这个逻辑结构为适应求解，即运算的需要，选择一个恰当的存储表示。前面的选择是面向问题，后面的选择是面向机器。这中间有一个“面向问题”的数据的逻辑结构向“面向机器”的数据的存储结构转换的问题，这正是数据结构所要研究的。

我们学习、研究数据结构的目的在于对大量的数据进行有效处理，合理地应用好计算机的两大资源——时间和空间。