21世纪 高等学校电子信息类专业规划教材

数据结构

张凤琴 主编
张水平 王蓉 副主编
万映辉 马礼举

清华大学出版社
http://www.tup.tsinghua.edu.cn
北京交通大学出版社
http://press.bjtu.edu.cn

21世纪高等学校电子信息类专业规划教材

数 据 结 构

张凤琴 主 编
张水平 王 蓉 副主编
万映辉 马礼举

清华大学出版社
北京交通大学出版社
·北京·

内 容 简 介

本教材是根据教育部制订的计算机科学与技术及相关专业的培养目标，突出对理论知识的应用和实际动手能力的培养，使基础理论的教学最终以应用为目的。本书在描述数据结构和算法时，程序结构清晰，可读性强，符合软件工程的规范要求。讲解的内容由浅入深，易于理解。文字表达简练清晰、通俗易懂。主要介绍了线性表、串、栈、队列、树和图等基本数据类型的基本概念，以及表示和算法实现，还介绍了静态、动态查找表的实现算法，各种内部排序的算法和文件的组织形式等。本书的算法均用类 C 语言描述。各章后均附有内容小结及习题，以帮助学生加深对所学知识的理解和掌握。

本书可作为高等院校计算机科学与技术专业及相关专业的本科教材，也可作为软件水平考试、计算机等级考试的参考书，对于从事软件应用开发的人员也是一本不可多得的参考书。

版权所有，翻印必究。举报电话：010－62782989　13501256678　13801310933
本书封面贴有清华大学出版社防伪标签，无标签者不得销售。
本书防伪标签采用特殊防伪技术，用户可通过在图案表面涂抹清水，图案消失，水干后图案复现；或将表面膜揭下，放在白纸上用彩笔涂抹，图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

数据结构/张凤琴主编. —北京：清华大学出版社；北京交通大学出版社，2005.7
(21 世纪高等学校电子信息类专业规划教材)
ISBN 7－81082－514－3

Ⅰ.数…　Ⅱ.张…　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(2005)第 029471 号

责任编辑：刘利平
出 版 者：清华大学出版社　邮编：100084　电话：010－62776969
　　北京交通大学出版社　邮编：100044　电话：010－51686414
印 刷 者：北京鑫海金澳胶印有限公司
发 行 者：新华书店总店北京发行所
开　本：185×260　印张：15.75　字数：376 千字
版　次：2005 年 7 月第 1 版　2005 年 7 月第 1 次印刷
书　号：ISBN 7－81082－514－3/TP·191
印　数：1～4 000 册　定价：22.00 元

本书如有质量问题，请向北京交通大学出版社质监组反映。对您的意见和批评，我们表示欢迎和感谢。
投诉电话：010－51686043，51686008；传真：010－62225406；E-mail：press@center.bjtu.edu.cn。

前　言

“数据结构”课程是计算机科学与技术、计算机通信及相关专业的专业基础课，本教材是依据教育部制订的关于计算机科学与技术及相关专业的培养目标及教学大纲的要求，以培养生产、建设、管理、服务等一线所需的技术应用型人才为目标，突出理论知识和应用能力的培养。通过这门课程的学习，使学生能够掌握执行速度快、占用空间少、可靠性高、可读性好的程序的编写方法与技巧，最终达到面对一个具体应用问题时，能选择最佳的逻辑结构、存储结构及实现算法，并能初步使用时间复杂度和空间复杂度来正确地评价算法的水平。

本书内容安排合理，详略得当；讲解时始终贯穿由浅入深、易于理解的宗旨，对于重要和复杂的概念，讲述时配有例题进行示范；在文字表达上力求简练清楚、概念清晰、通俗易懂；实现的算法具体，易于调试；描述的数据结构和算法结构清晰、可读性高、符合软件工程的规范，学生的学习过程也就是复杂程序的设计过程。本书的算法使用类 C 语言进行描述，配有相关的解释，以帮助读者理解。

全书共分 10 章。第 1 章介绍数据结构基本概念、数据的逻辑结构、数据的存储结构、数据的运算、算法的评估；第 2 章介绍线性表基本概念、各种存储结构及应用；第 3 章介绍栈与队列基本概念、各种存储结构及应用；第 4 章介绍串的基本概念、各种存储结构及应用；第 5 章介绍数组和广义表的基本概念、存储结构及应用；第 6 章介绍树和二叉树基本概念、各种存储结构及遍历、线索化二叉树、哈夫曼树及其应用；第 7 章介绍图的基本概念、各种存储结构及遍历、最小生成树、拓扑排序、关键路径、最短路径；第 8 章介绍查找的基本概念、静态查找表及动态查找表的实现算法、哈希表的构造与查找算法；第 9 章介绍各种内部排序的算法、算法比较；第 10 章介绍文件的基本概念、不同文件结构的应用。每一章后面均附有小结与习题，以帮助学生加深对内容的理解，巩固知识，提高学习效率。

本书的第 1 章和第 9 章由张凤琴编写；第 2,3,4 章由王蓉编写；第 5 章和第 7 章由杜炜编写；第 6 章由张水平编写；第 8 章由万映辉编写；第 10 章由马礼举编写。张凤琴和张水平制定了编写大纲，最后由张凤琴和王蓉对全文进行通审和定稿。

本教材由高等院校具有丰富教学和开发经验的一线教师和科研人员精心设计和撰写，将充分体现实践要求与教学目标统一的原则。在本书的编写过程中，西北工业大学赵正文教授，以及空军工程大学刘守义教授、殷肖川副教授和张月玲副教授对本书的编写提出了宝贵的意见；杨利和陈爱网对本书的代码进行调试；朱涛和王东对本书的图进行加工处理。编者在此对他们表示衷心的感谢。

由于作者水平有限，书中错误和疏漏之处敬请读者指正。欢迎读者用 E-mail 与作者联系，作者邮箱：fengqin _ zhang@126.com。

编　者

2005 年 7 月

目　录

试读结束：需要全本请在线购买：　www.ertongbook.com

第1章 概 述

21世纪是一个信息业高速发展的时代，计算机的应用已深入到人类社会的各个领域，其应用范围也由最初的科学计算，延伸到数据处理、管理和实时控制等领域。与此同时，计算机加工处理的对象也从纯粹的数值发展到字符、表格、声音、图形、图像等各种复杂的而且具有一定结构的数据。因此，在进行程序设计的过程中，除了应用一些程序设计的技巧和方法外，必须研究数据的特性和数据之间存在的内在关系，才能设计出优质的程序。本章主要介绍数据结构有关的概念和术语，如数据、数据元素、逻辑结构、存储结构、数据处理、数据结构、算法设计等基本概念，以及如何评价一个算法。

1.1 数据结构的基本概念

“数据结构”在计算机科学中是一门综合性的专业基础课，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，其内容不仅是一般程序设计（特别是非数值性程序设计）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的重要基础。使用“数据结构”在处理实际问题的过程中涉及到许多概念，本节主要介绍数据结构的定义、基本概念和术语。

1.1.1 数据结构的定义

在进行科学计算时，需要经过的几个步骤是：首先要从具体问题中抽象出一个适当的数学模型，然后设计一个解此数学模型的算法（Algorithm），最后编出程序，进行测试、修改直至得到最终解答。在解决问题的过程中，寻求数学模型的实质是分析问题，即从问题中提取操作的对象，并找出这些操作对象之间相互的关系，然后用数学的语言加以描述。例如求一元二次方程的根；天气预报的计算；水库大坝的应力计算等。但在非数值计算时，通常无法使用数学模型来描述，如查询图书大厦的书目、各种图书的数量、销售情况等问题。下面先讨论3个例子。

【例1.1】 学生的基本资料如表1-1、表1-2所示。

表1-1的学生基本情况中，如果学生的学号、姓名、课程名等是随机排列的，那么，要查找某个班级的某个学生的基本情况，则需要从表中第一个学号开始，逐个与给定学号、姓名或班级进行比较，直到找到相应的学生。这种方法简单，但非常费时，效率低。如果对学生基本情况进行适当地整理，例如按学生的班级进行排列，同时构造一个索引表用来登记每个班级的学生在学生基本情况表中的起始位置，这样查找学生成绩时，可以从该学生所在班级的位置开始查找，而不必查看其他部分。显然，在此基础上编写的查找程序执行效率高。如果希望了解某学生“高等数学”成绩在本班级的排名情况，则首先需要查询本班级同学的学号，然后通过学号查找“高等数学”成绩，最后按照“高等数学”成绩排序。若利用计算机来完成自动检索，则需要在学生基本情况中列出学生的学号、姓名、性别、年龄、民族、班级、专业

等信息，而在学生成绩表中只需列出学生的学号、课程名、考试的学期以及成绩等信息，如表1-2所示。这就是由两张表构成的学生情况计算机查询的数据模型。由此可见，查找算法的设计，完全依赖于学生基本资料的组织方式，这就是一个数据结构问题。不同的数据结构可以有不同的算法，数据结构直接影响算法的选择和算法的效率。

表1-1 学生基本情况

学号	姓名	性别	年龄	民族	班级	专业
0211014	李敖	男	20	汉族	020101	计算机应用
0211015	李鸿	女	20	汉族	020101	计算机应用
0211016	赵均	男	19	汉族	020101	计算机应用
0301005	张刚	男	18	回族	030102	软件
……						

表1-2 学生成绩

学号	课程名	学期	成绩
……			
0211015	高等数学	1	81
0211015	英语	1	68
0211016	线性代数	3	75
……			

在这张表中，每一行描述了一个人(对象)的有关信息，表中对象之间的关系是顺序的，是一种最简单的线性关系，其数学模型是线性的数据结构。表中对象之间的关系直接反映了对象在计算机中如何进行存储表示。另外，在这张表中若有新生加入时要增加对象，若有学生退学时要删除相应对象。因此，从计算机管理学生资料表的问题抽象出来的数学模型应含有对表的插入、删除等运算。

【例1.2】 大学的机构管理形式，如图1-1所示。

一个大学的校部下设6个学院和一些直属单位，他们有通信工程学院、计算机学院、软件学院、管理学院、电子工程学院、研究生院、教务处、行政处和后勤处等。每个学院之下设立院机关和系，每个系之下设置多个教研室，在此之下还可细化。如研究生院有统考的研究生、在职研究生、定向研究生等，每类研究生可化分为不同年级的研究生。教务处之下设立教学计划办公室、考务办公室、学生管理办公室等；行政处之下设立党政办公室、人力资源办公室、招生就业办公室等；后勤处之下设立财务办公室、水电暖办公室、维修办公室、学生公寓办公室、食堂和医院等。这种结构为一个倒立的“树”，称该数据模型为树。如果想找某学校的“人力资源办公室”，则应该从校部开始，查找其直属单位的“行政处”，最后查找到该学校的“人力资源办公室”。这里的校部称为“树根”，最基层的单位称为“叶子”。在处理该类问题时，利用“树”来表示它们之间的逻辑关系，因此可把对该类问题的操作转换成对一个树形结构进行增加、删除、查找等运算。

【例1.3】 城市间的交通网络，如图1-2所示。

在这个图中，圆圈表示城市，边表示城市间的交通。在这个交通网络中，每一个城市就

试读结束：需要全本请在线购买： www.ertongbook.com

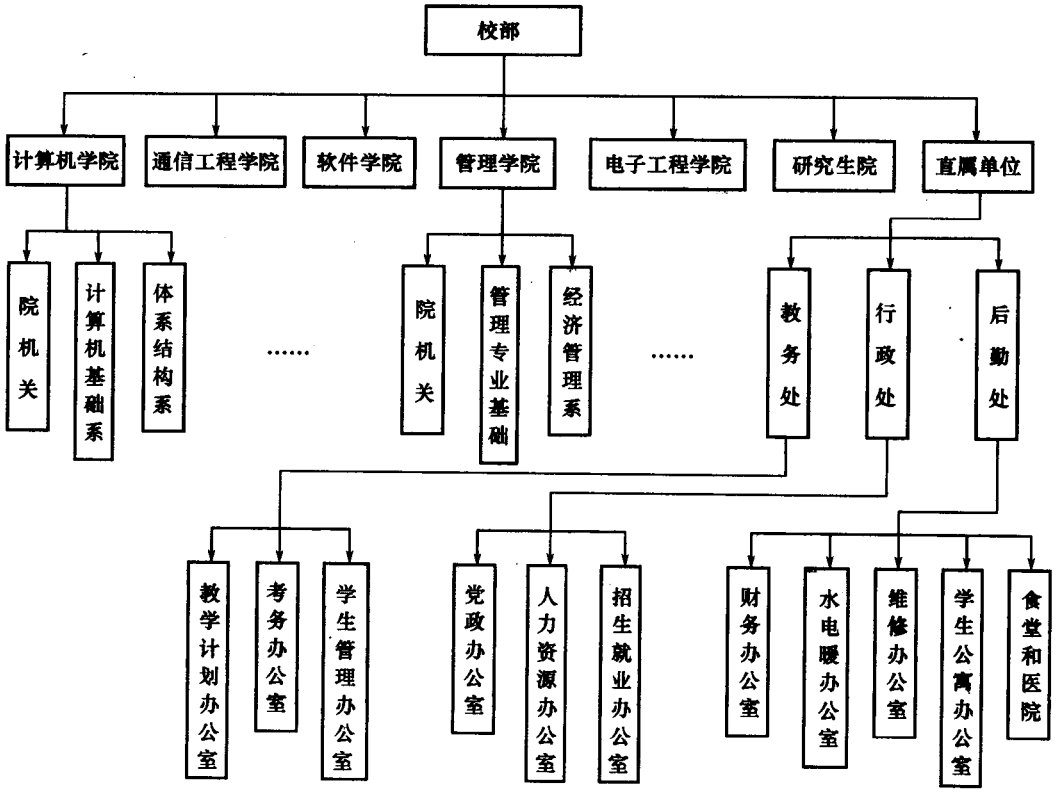


图 1-1 学校的机构

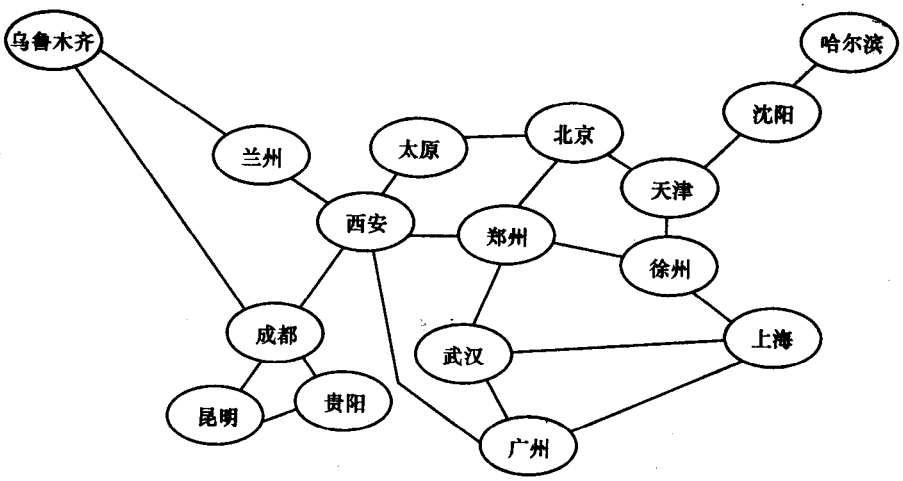


图 1-2 城市交通网

是一个顶点,城市与城市之间有直接的交通工具,如汽车、火车或飞机等,从一个城市到另一个城市可以通过多条路径抵达,这些路径形成一个“图”,顶点之间的逻辑关系不是顺序关系,故称这种数据模型为“图”。例如某乘客想从西安去北京,可以乘坐汽车、火车或飞机,如

果该乘客希望在路上花费的时间较少，则选乘飞机；如果该乘客希望在路上边走边欣赏风景，则选乘汽车、火车，此时需要选择中间经过太原，还是郑州，这里同样有一个时间和费用的问题。对于相邻城市的距离、任两个城市间的最短路径、所用的时间和费用等问题，可转换成对图的运算。

综合以上3个例子可见，描述这样一类问题的数学模型不再是数值方程，而是诸如表、树、图等的非数值性的数据结构及其运算。因此，简单的说，数据结构就是一门研究非数值程序设计中计算机操作的对象以及它们之间的关系和操作等的学科，即含有3方面的内容：数据元素间的逻辑结构、存储结构及其数据的操作。

1.1.2 基本概念和术语

数据结构所研究的内容涉及到编码理论、存储装置、数据的组织、程序设计等多门学科，因此，在讲解内容之前，先介绍基本概念和术语的确切含义。

1. 数据(Data)

数据是描述客观事物的数、字符以及所有能够输入到计算机中并被计算机程序处理的符号的集合，是计算机程序加工的“原料”。例如，在数值计算中所使用的数据是整数和实数；文本编辑程序中使用的数据是字符串。随着计算机技术的发展和计算机应用的普及，数据的含义也变得广泛了，现在，图像、声音等也可以经过一定转化变换成为数字化的信息数据，利用计算机进行加工和处理，所以图像和声音也属于数据的范畴。

2. 数据元素(Data Element)和数据项(Data Item)

数据元素是数据的基本单位，表现为数据这个集合中的一个个体。在计算机程序中通常把数据元素作为一个整体进行考虑和处理，通常一个数据元素由一个或若干个数据项(Data Item)组成。例如，在例1.1中，一个学生的信息为一个数据元素，它由学号、姓名、性别、年龄、班级、专业等数据项组成；在例1.2中，数据元素为一个单位，它可由单位编号、单位名称、办公地址、联系电话等数据项组成；在例1.3中，数据元素为一个城市，它可由城市编号、城市名称、城市位置、名胜古迹等数据项组成。

数据元素有时也称为结点或记录。

数据项是具有独立含义的最小标识单位。例如，在学生管理系统中的学生是一个数据元素，组成它的数据项(如姓名、年龄、性别等)均为不可分割的最小单位。

3. 数据对象(Data Object)

数据对象是具有相同特性的数据元素的集合，是数据的一个子集。例如，学生管理系统中的数据对象就是该学校的全体学生。

4. 数据类型(Data Type)和抽象数据类型(Abstract Data Type)

数据类型是指程序设计语言中各变量所属的数据种类。数据类型是高级程序设计语言中的一个基本概念，它和数据结构的概念密切相关。数据类型按“值”的不同特性可分为原子类型和结构类型。原子类型的值是不可分解的，例如C语言中的整型、实型、字符型、枚举型、指针类型和空类型；结构类型的值是由若干个成分(原子类型或结构类型)按某种结构组成，因此它是可以分解的，例如图1-1中的学生类型，它是由学号、姓名、性别、年龄、班级、专业等成分组成。

一方面,在程序设计语言中,每一个数据都属于某种数据类型。类型明显或隐含地规定了数据的取值范围、存储方式以及允许进行的运算。可以认为,数据类型是在程序设计中已经实现了的数据结构。另一方面,在程序设计过程中,当需要引入某种新的数据结构时,总是借助编程语言所提供的数据类型来描述数据的存储结构。

抽象数据类型(简称 ADT)通常是指对数据的某种抽象,它定义数据的取值范围及其结构形式,是对数据进行操作的集合。抽象数据类型描述数据的构造及使用,并不注重其在程序中如何实现。抽象数据类型通过一种特定的数据结构在程序的某个部分得以实现,而在设计使用抽象数据类型的那部分程序时,不关心数据结构的具体实现,而仅关心这个数据类型上的操作。因此,在思考一个复杂的程序时,首先应考虑能否将它的处理对象抽象为一种数据结构,否则很难理解或者实现它。

在定义抽象数据类型时,通常需要定义数据对象、数据关系和基本操作。本书中对抽象数据类型的定义则遵循该规则。

5. 逻辑结构(Logical Structure)

结点和结点之间的逻辑关系称为数据的逻辑结构。在表 1-1 中,各结点之间在逻辑上有一种线性关系,即学号为"0211015"的学生结点前面紧相邻的是学号为"0211014"的学生,后面紧邻的学号为"0211016"的学生,以此类推,最终指定了各个结点在表中的排列顺序。根据这种线性关系,可以看出表中第一个学生到最后一个学生的情况。

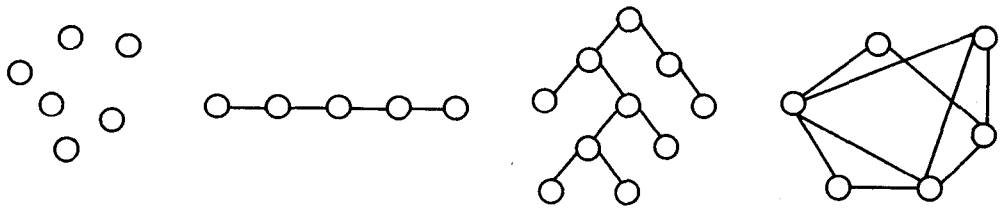
6. 存储结构(Storage Structure)

数据在计算机中的存储表示称为数据的存储结构,又称为数据的物理结构。例如对于表 1-1 所示的表格数据,在计算机中可以有多种存储表示,既可以表示成数组存放在内存中;也可以表示成文件存放在磁盘上等。

7. 数据结构(Data Structure)

数据结构是研究数据元素之间抽象化的相互关系和这种关系在计算机中的存储表示(即数据的逻辑结构和物理结构),并对这种结构定义相适应的运算,设计出相应的算法,而且确保经过这些运算后所得到的新结构仍然是原来的结构类型。

根据数据元素间的关系不同,将数据结构划分为 4 种基本结构:集合、线性结构、树形结构和图状结构,如图 1-3 所示。集合中的数据元素同属于一个集合;线性结构中的数据元素存在着一个对一个的关系;树形结构中的数据元素存在着一个对多个的关系;图状结构中的数据元素存在着多个对多个的关系。由于"集合"的各数据元素间的关系极为松散,因此可用其他结构来表示。



(a) 集合 (b) 线性结构 (c) 树状结构 (d) 图状结构

图 1-3 基本结构

为了叙述上的方便和避免产生混淆，通常把数据的逻辑结构统称为数据结构，把数据的物理结构统称为存储结构。

8. 数据处理(Data Process)

早期的计算机主要用于数值计算，20 世纪 80 年代以后，计算机主要用于非数值性计算。所以说，数据处理是指对数据进行科学计算、查找、插入、删除、合并、排序和统计等操作的过程。

9. 算法(Algorithm)

算法是指解决问题的一种方法或一个过程。算法需要用一种语言来描述，同时，算法又有各种描述方法以满足不同的需求。例如，一个需要在计算机上运行的程序(程序也是算法)必须是按照语法用机器语言、汇编语言或高级程序语言编写，而一个为了便于人们阅读和交流的算法则可以用伪码语言或框图等其他形式来描述。由于本书中讨论的算法主要是为读者阅读的，且能容易地转换成用高级语言书写的程序，因此采用类 C 语言描述。

一般说来，在数值算法中主要进行算术运算，而在非数值算法中主要进行比较和逻辑运算。另一方面，特定的问题可能是递归的，也可能是非递归的，因而解决它们的算法就有递归算法和非递归算法之分。从理论上讲，任何递归算法都可以通过循环、堆栈等技术转化为非递归算法。

1.2 数据的逻辑结构

数据的逻辑结构指的是数据元素间的逻辑关系，又称数据结构。数据的逻辑结构又分为线性结构和非线性结构。线性结构的逻辑特征是有且仅有一个开始元素和一个终结元素，除第一个元素外，其他的元素有且仅有一个与它相邻且在它前面的元素(称为直接前驱)，除最后一个元素外，其他元素都有且仅有一个与它相邻且在它后面的元素(称为直接后继)。非线性结构的逻辑特征是一个结点可能有多个直接前趋和多个直接后继，例如前面提到的树形结构，它有且仅有一个没有前趋结点的结点，称之为根结点。其他结点都仅有一个前趋结点，但允许有零个或多个后继结点，对于无后继结点的结点，称之为叶子结点。从根结点到任一非根结点，都有且仅有一条路径。

数据的逻辑结构可以用一个二元组来表示。

$$\text{Data\_Structure} = (D, R)$$

其中 D 是结点的有穷集合，即 D 是由有限个结点所构成的集合；R 是 D 上的关系的有穷集合，即 R 是由有限个关系所构成的集合。

对于结点的值可以是非结构类型和结构类型。例如结点的值可以是整型或字符型，也可以是例 1.1 的结构类型，采用 C 语言的描述如下：

```
struct student{
    char    Stuid [6];          /* 学号 */
    char    Name [10];          /* 姓名 */
    char    Sex [2];            /* 性别 */
    int     Age;                /* 年龄 */
    char    People [10];        /* 民族 */
```

```
    char    Classes[6];          /*班级*/
    char    Depno[20];           /*专业*/
};
```

对于结点上的关系使用有序偶来表示，即$<a_1,a_2>$，其中a_1,a_2是结点集中的数据元素，且它们之间具有直接前驱和直接后继的关系。

下面给出两个数据逻辑结构的例子，为了描述清楚，用图来表示数据的逻辑结构。其中长方形框表示结点，长方形框之间用有向线段连接，表示结点之间的前驱/后继关系。

【例1.4】 数据逻辑结构LS1=(D,R)，其中

$D=\{d_1,d_2,\cdots,d_9\}$，$R=\{r\}$，

$r=\{<d_1,d_2>,<d_1,d_3>,<d_1,d_4>,<d_1,d_7>,<d_1,d_8>,<d_2,d_5>,$
$<d_2,d_6>,<d_7,d_9>\}$

用图1-4表示LS1的逻辑结构。其中d_1为开始结点(根结点)，d_3,d_4,d_5,d_6,d_8,d_9为终端结点(叶子结点)。

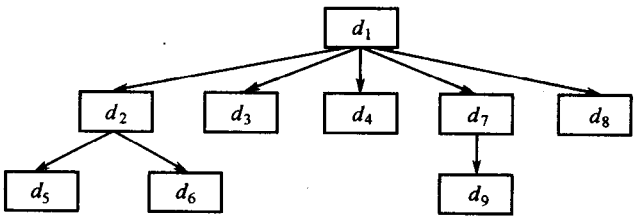


图1-4 LS1的逻辑结构

【例1.5】 数据逻辑结构LS2=(D,R)，其中

$D=\{d_1,d_2,\cdots,d_{10}\}$，$R=\{r_1,r_2\}$

$r_1=\{<d_1,d_2>,<d_1,d_8>,<d_2,d_3>,<d_2,d_7>,<d_3,d_4>,$
$<d_3,d_5>,<d_8,d_9>,<d_8,d_{10}>,<d_9,d_6>\}$

$r_2=\{<d_2,d_3>,<d_4,d_2>,<d_6,d_3>,<d_6,d_1>,<d_{10},d_4>\}$

用图1-5表示LS2的逻辑结构，其中实线表示关系r_1，虚线表示关系r_2。

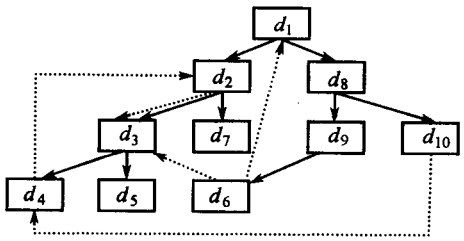


图1-5 LS2的逻辑结构

1.3 数据的存储结构

数据元素及其关系在计算机中的存储表示称为数据的物理映像(物理结构)，又称为存

储结构。在数据的存储结构中,应包含数据元素和前驱与后继的关系(可以隐含)两类映像。按照不同的存储方式,数据存储结构可分为顺序存储结构和非顺序存储结构。

在计算机中表示信息的最小单位是二进制数的一位,叫做位(bit)。在计算机中,可以用一个由若干个位组合形成的一个位串表示一个数据元素,通常称这个位串为元素或结点。例如,用 8 位二进制表示一个字符;用 16 位二进制表示一个整数等。

计算机的存储器(主存)是由有限多个存储单元组成的,每个存储单元有惟一的地址,各存储单元的地址是连续编码的,每个存储单元 Z 都有惟一的后继单元 Z',Z 和 Z' 称为相邻单元。一片相邻的存储单元的整体叫做存储区域,记作 M。在不产生混淆的情况下,Z 可以用来表示一个存储单元,也可以用来表示该存储单元的地址。把各个结点的值存放在存储器中连续的存储单元内称为顺序存储结构,否则称为非顺序存储结构。

设有逻辑结构 $B=(D,R)$,要把 B 存储在计算机中,必须建立一个从 B 到 M 的单元的映像 $S:B\rightarrow M$,即对于每一个 $d\in D$ 都有惟一的 $Z\in M$,使得 $S(d)=Z$,Z 为 D 结点所占存储空间中的开始单元。同时这个映像应具有显式或隐式地体现关系 R 的能力。

由于计算机的存储区域总是有限的,所以如何合理地使用它,使得有限的存储区域发挥最大的作用,这是存储管理问题,在以后章节中将对这个问题进行讨论。

我们用 LOC(d)表示结点 d 对应的存储单元的地址。下面介绍 4 种基本存储映像方法。

1. 顺序方法

这种方法主要用于线性的数据结构,它把逻辑上相邻的结点存储在物理上相邻的存储单元里,结点之间的关系由存储单元的邻接关系来体现。即如果结点 d 所占存储空间的第一个单元为 Z,即 LOC(d)=Z,而最后一个单元为 Z_1,那么 Z_1 的后继单元 Z' 就是 d 的后继 d' 的第一个存储单元。例如,如果线性结构的数据元素为{1,2,3,4,5,6},则在 M 中的表示如图 1-6 所示。

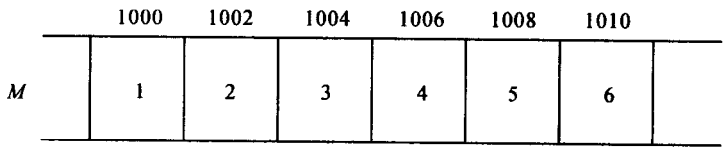


图 1-6 顺序存储结构示意图

在图 1-6 中,每个存储结点只包含一个数据元素,假设整数占据 2 个字节,第一个数据元素的存储单元为 1000 和 1001,那么第二个数据元素的存储单元为 1002 和 1003,依此类推,即所有存储结点的次序是连续的,且中间没有空出存储单元。如果在存储结点的存储单元中间存在空的存储单元,则该存储结构便不是一个顺序存储结构。

另外对非线性的数据结构也可以采用局部线性化的方法实现顺序存储。例如,在树形结构中可以把结点按某种规则排成序列,用顺序存储方法把结点内部的信息稠密地存放在一起,而对结点之间的关系采用其他的存放方法。

2. 链式方法

这种方法是给结点附加指针域。即,将结点所占的存储单元分为两部分:一部分存放结点本身的信息,称为数据域;另一部分存放此结点的后继结点所对应的存储单元的地址,称

试读结束:需要全本请在线购买: www.ertongbook.com