

21世纪网络平台大学计算机系列教材



SHU JU JIE GOU 数据结构

SHU JU JIE GOU

王少波 主编



科学出版社
www.sciencep.com

· 21 世纪网络平台大学计算机系列教材 ·

数 据 结 构

王少波 主编

科 学 出 版 社

北 京

内 容 简 介

本书是按高等院校计算机专业及信息管理专业本科四年制教学计划数据结构课程教学大纲要求编写的教材。全书共分十章,分别介绍了线性表、栈与队列、串、树、图、数组、内部排序、查找、文件等基本数据结构、算法及其应用例题。

本书可作为计算机专业、信息管理专业及相关专业的教材,还可作为计算机科技工作者及其有关专业人员的参考书。

图书在版编目(CIP)数据

数据结构/王少波主编. - 北京:科学出版社,2004

(21世纪网络平台大学计算机系列教材)

ISBN 7-03-013947-x

I.数… II.王… 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(2004)第072618号

责任编辑:冯贵层

责任印制:高 嵘/封面设计:张 琴

科学出版社出版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

武汉大学出版社印刷总厂印刷

科学出版社发行 各地新华书店经销

*

2004年8月第 一 版 开本:787×1092 1/16

2004年8月第一次印刷 印张:17 1/4

印数:1-6 000 字数:422 000

定价:25.80元

(如有印装质量问题,我社负责调换)

前 言

数据结构发展至今,已成为一门比较成熟的课程。它是计算机系统软件和应用软件研制者的必修课程。数据结构和算法的研究是计算机基础性研究之一,掌握这个领域的知识对于利用计算机资源高效地开发计算机程序是非常必要的。

数据结构理论的应用范围已经深入到编译系统、操作系统、数据库、人工智能、信息科学、系统工程、计算机辅助设计及其他信息管理的应用。数据结构主要解决非数值计算应用问题。

从理论上讲,数据结构的概念严谨、抽象;每种数据结构类型描述层次清晰可见——概念层、逻辑定义层、物理存储层、运算实现层;每种数据结构类型描述反映了实现问题的思想、实现的前提以及不同实现方式的特点和优劣;数据结构描述的内容看上去如同程序,但不是程序,它是程序设计思想的抽象化、一般化,它不依赖于某种物理设备,甚至某种语言系统,受教育者通过数据结构课程的学习,不仅能从中学到专业知识,而且能学到一种思维方式。

从实践上讲,数据结构是建立在抽象化描述基础之上的实践性理论,这门学科只有赋予实践的内容才具有完备性,具体化是该学科的又一特点。在计算机系统中,全面体现着数据结构所在,系统框架结构的构建、程序实现的精巧化都熔入了数据结构的理论思想和技术。

本书是按高等院校计算机专业及信息管理专业本科四年制教学计划数据结构课程教学大纲要求编写的教材。全书共分十章,介绍了线性表、栈和队列、串、树、图、数组、内部分类、查找、文件等基本数据结构、算法及其应用例题。

本书叙述了各种基本数据结构的概念,包括数据结构的逻辑定义、物理实现及其相应运算,并举例说明怎样用这些抽象的概念来解决实际问题。叙述时回避了复杂的数学定义与推导,力求通俗易懂。读者通过本书的学习不仅能正确地掌握数据结构的基本理论,并能运用这些理论来解决实际问题。

本书系作者集多年来从事计算机软件设计实践及讲授数据结构课程的体会,并参考分析了国内外数据结构书籍文献编写而成。本书采用广泛使用的 C++ 语言描述算法,并进行了适当的算法复杂性分析。

数据结构不但是一门理论性很强的课程,同时又是一门实践性很强的课程,在每一章的最后都安排了适当的习题,供读者练习。全书还给出了每章应进行的上机实践内容、目的及要求。

本书由王少波主编,范爱萍、刘晋萍任副主编,屈振新参加编写。其中第一、二、三、五章由王少波编写;第八、九章由范爱萍编写;第四、七章由刘晋萍编写;第六、十章由屈振新编写。全书由王少波负责统稿和定稿。

在成书过程中,编者参考了有关书籍,在此向这些书籍的作者一并表示感谢。

由于编者水平有限,书中错误与疏漏之处恳请读者不吝指教。

编者

2004年8月

目 录

第一章 绪论	1
1.1 什么是数据结构	1
1.1.1 数据结构相关事例	2
1.1.2 数据结构的定义	4
1.2 数据结构的相关概念	5
1.2.1 数据和信息	5
1.2.2 数据元素	5
1.2.3 结构类型	6
1.2.4 静态存储空间分配和动态存储空间分配	9
1.3 数据类型、抽象数据类型和数据结构	10
1.4 算法及算法分析、算法描述	13
1.4.1 算法和程序	13
1.4.2 程序性能和算法效率	14
1.4.3 算法分析	15
1.4.4 算法描述	18
习题一	19
第二章 线性表	21
2.1 线性表的定义	21
2.1.1 线性表的逻辑结构	21
2.1.2 线性表的抽象数据类型	21
2.2 线性表的顺序存储及操作	22
2.2.1 线性表顺序存储	22
2.2.2 线性表顺序存储结构下的操作	24
2.3 简单链表存储结构及操作	27
2.3.1 简单链表的存储	27
2.3.2 简单链表的操作	29
2.4 双向链表	34
2.4.1 双向链表的存储	34
2.4.2 双向链表的操作	35
2.5 单向循环链表和双向循环链表	37
2.5.1 单向循环链表的存储	37
2.5.2 双向循环链表的存储	38
2.6 模拟指针方式构造简单链表	39
2.6.1 模拟链表的存储	39
2.6.2 模拟链表的操作	41

2.7 多重链表	43
2.8 链表应用	45
2.8.1 结点移至表首运算	45
2.8.2 链表的逆向运算	46
2.8.3 多项式的相加运算	47
2.8.4 十字链表结构的应用	49
2.8.5 一个较复杂的机票售票系统的数据结构方案	50
习题二	52
第三章 栈与队列	54
3.1 堆栈的定义	54
3.1.1 堆栈的逻辑结构	54
3.1.2 堆栈的抽象数据类型	54
3.2 堆栈的顺序存储及操作	55
3.2.1 堆栈顺序存储	55
3.2.2 堆栈顺序存储结构下的操作	56
3.3 堆栈的链式存储及操作	58
3.3.1 堆栈的链式存储	58
3.3.2 链式栈的操作	59
3.4 多个栈共享邻接空间	60
3.5 堆栈的应用	61
3.5.1 检验表达式中括号的匹配	61
3.5.2 表达式的求值	63
3.5.3 背包问题求解	64
3.5.4 地图四染色问题求解	65
3.6 队列的定义	68
3.6.1 队列的逻辑结构	68
3.6.2 队列的抽象数据类型	69
3.7 队列的顺序存储及操作	70
3.7.1 队列顺序存储	70
3.7.2 队列顺序存储结构下的操作	73
3.8 队列的链式存储及操作	75
3.8.1 队列的链式存储	75
3.8.2 链式队列的操作	76
3.9 队列的应用	77
3.9.1 列车重排	77
3.9.2 投资组合问题	80
习题三	82
第四章 串	84
4.1 串的定义	84
4.1.1 串的逻辑结构	84

4.1.2 串的抽象数据类型	84
4.2 串的表示和实现	85
4.2.1 串的静态顺序存储结构	85
4.2.2 串的动态顺序存储结构	87
4.2.3 串的链式存储结构	89
4.3 串的模式匹配算法 IndexStr(S1,S2,p)	90
4.3.1 普通模式匹配算法 IndexStr(S1,S2,p)	91
4.3.2 改进的模式匹配算法 KMPIndexStr(S1,S2,p)	92
习题四	95
第五章 树	97
5.1 树、森林的概念	97
5.1.1 树的定义	97
5.1.2 树的术语	97
5.2 二叉树定义及性质	98
5.2.1 二叉树的定义	98
5.2.2 二叉树的性质	100
5.2.3 二叉树的抽象数据类型	102
5.3 二叉树的存储结构	103
5.3.1 二叉树的顺序存储概念	103
5.3.2 二叉树的链式存储结构	104
5.4 二叉树链式存储结构下的操作	105
5.4.1 二叉树的操作概念	105
5.4.2 二叉树的前序、中序、后序遍历操作	106
5.4.3 二叉树的层次遍历操作	112
5.4.4 二叉树的其他操作	114
5.5 线索树	116
5.5.1 线索树的概念	116
5.5.2 二叉线索树的操作	119
5.6 一般树的表示和遍历	126
5.6.1 一般树的二叉链表示以及它与二叉树的关系	126
5.6.2 二叉树、一般树及森林的关系	127
5.6.3 一般树的遍历概念	128
5.6.4 一般树的运算	129
5.7 树的应用	130
5.7.1 分类二叉树	130
5.7.2 堆树	135
5.7.3 树的路径长度和哈夫曼树(Huffman)	142
5.7.4 判定树	148
习题五	149
第六章 图	151

6.1 图的概念	151
6.1.1 图的定义	151
6.1.2 图的术语	151
6.1.3 图的抽象数据类型	153
6.2 图的存储结构	153
6.2.1 邻接矩阵表示法	153
6.2.2 邻接表表示法	156
6.3 图的遍历	158
6.3.1 深度优先搜索遍历	159
6.3.2 宽度优先搜索遍历	161
6.4 最小生成树	162
6.4.1 生成树	162
6.4.2 最小代价生成树	163
6.5 最短路径	166
6.5.1 单源最短路径	167
6.5.2 任意两个顶点之间的路径	168
6.6 拓扑排序	170
6.6.1 AOV 网	170
6.6.2 拓扑排序	171
6.7 关键路径	173
6.7.1 AOE 的概念	173
6.7.2 关键路径的概念	174
6.7.3 关键路径的算法	174
习题六	177
第七章 数组	179
7.1 数组的定义	179
7.1.1 数组的逻辑结构	180
7.1.2 数组的抽象数据类型	181
7.2 数组的顺序表示及运算	182
7.2.1 数组的顺序存储结构	182
7.2.2 数组顺序存储结构描述	184
7.2.3 数组顺序存储结构下的操作	186
7.3 矩阵的存储及操作	187
7.3.1 矩阵的定义及操作	187
7.3.2 矩阵的顺序存储	187
7.3.3 特殊矩阵的压缩存储及操作	187
7.3.4 稀疏矩阵的压缩存储及操作	189
习题七	202
第八章 内部排序	204
8.1 排序的基本概念	204

8.2 待排序数据对象的存储结构	206
8.3 插入排序	206
8.3.1 直接插入排序	206
8.3.2 折半插入排序	208
8.3.3 希尔排序	209
8.4 交换排序	211
8.4.1 冒泡排序	211
8.4.2 快速排序	212
8.5 选择排序	216
8.5.1 直接选择排序	216
8.5.2 堆排序	217
8.6 归并排序	218
8.7 基数排序	221
8.7.1 用二维数组表示桶	222
8.7.2 用链式存储结构实现桶	224
习题八	226
第九章 查找	227
9.1 查找的概念	227
9.2 静态查找技术	228
9.2.1 顺序查找	229
9.2.2 二分查找	230
9.2.3 分块查找	233
9.3 动态查找技术	236
9.3.1 B-树的定义和表示	236
9.3.2 B-树的查找	237
9.3.3 B-树的插入	238
9.3.4 B-树的删除	241
9.4 哈希表的查找	242
9.4.1 基本概念	242
9.4.2 构造哈希函数的方法	243
9.4.3 哈希冲突的解决方法	245
9.4.4 哈希表的查找	247
9.4.5 哈希算法	247
习题九	249
第十章 文件	251
10.1 外部存储设备	251
10.1.1 磁带	251
10.1.2 磁盘	252
10.1.3 光盘	253
10.1.4 闪存	253

10.2 基本概念.....	254
10.3 顺序文件.....	255
10.4 索引文件.....	256
10.5 索引顺序文件.....	257
10.6 直接存取文件.....	259
10.7 倒排文件.....	259
习题十.....	260
附录 实践内容及要求.....	261

第一章 绪 论

计算机科学是一门研究信息表示和处理的科学。信息表示包括组成信息的元素之间的相互关系（逻辑顺序）和信息元素在计算机中的存储方式（物理顺序）；信息处理是根据解决实际问题的需要对信息进行加工计算的过程。

在计算机领域中，一般所讨论的“计算”有别于数学概念中的一般计算，通常将计算又称为“运算”或“操作”。运算或操作的内涵不仅包括传统意义上的四则运算和各种函数运算（公式化运算），而且还包括数据存取、插入、删除、查找、排序和遍历等运算。

从第一台电子计算机问世以来，计算机的应用主要包括两个方面：数值计算和非数值计算。计算机发展的初期，计算机主要为数值计算服务，其特点是计算过程复杂，数据类型相对简单，数据量相对较少；随着计算机的应用范围深入到各个领域，应用方面也不再限于数值计算，更多地表现为非数值计算应用。非数值计算应用的特点表现为计算过程相对简单，数据类型相对复杂，数据的组织排列从某种意义上决定着非数值计算应用的有效性。

数值计算主要是指对一个或一组数据进行较复杂的四则运算、函数运算或迭代运算等。运算过程表现为数据处理的深入性，一般运算的原始数据对象较少。数据类型一般是数值型数据（整型、浮点型）。数值计算的程序设计主要围绕程序设计技巧，是典型的以程序为中心的设计过程。

非数值计算主要是指对类型较为复杂的大量数据进行内在联系的分析，根据处理的需要，合理地将数据按一定结构顺序进行组织存储，并完成对数据处理的程序描述。在非数值计算中，一般所处理的数据对象的类型较为复杂，通常是描述一个实体的若干个属性值的集合（结构类型或记录类型）；另外，数据是大量的。由于数据的大量性，所以移动全部或部分数据将消耗大量的时间或空间；还有，对数据进行较复杂的四则运算、函数运算等相对较少，较多的是对数据进行“管理”运算，如存取、查找、排序、插入、删除、更新等。

解决非数值计算问题，仅仅依赖程序设计的技巧已经无法达到目的，必须对这些被加工数据的组织形式加以研究，找出最佳的数据组织形式，并与好的程序设计技巧相配合，才能达到提高效率的目的。所以，非数值计算问题是以复杂的数据为中心，研究数据的合理组织形式，并设计出基于合理数据组织结构的高效程序。

1.1 什么是数据结构

数据结构是随着计算机科学的发展而建立起来的围绕非数值计算问题的一门学问，而非数值计算是先于计算机科学存在的学问。在处理非数值计算问题时，首先要建立问题的数据模型，然后设计相应的算法。数据模型包含数据的组成结构、数据间的关联方式以及对数据实施相应运算后数据组成结构的完整性。数据组成结构的完整性是指，不因对数据运算而改变数据模型的性质，运算方法本身是在保证数据组成结构完整性的前提下以相同规律进行的。

1.1.1 数据结构相关事例

为了说明什么是数据结构,我们先讨论现实生活中的几个例子。

问题一:电话号码簿的使用及字典的使用。

当用户拿起一本厚厚的电话号码簿,查找自己需要的单位或个人的电话号码时,一定是从电话号码簿的分类目录开始,查找相应的大类别,然后根据所查找到的大类别后面指定的页码,翻到大类别的起始页,再从特定的大类别中查找小类别,从检索到的小类别下面顺序地找到用户所要的单位或个人的电话号码(如图 1.1.1)。

党政机关	7	7	省委	55	55	省委办公厅一处	88060001
大学	12		市委	127		省委办公厅二处	88060002
企业	25		区委	224		.	
.			.			.	
.			.		127	市委办公一处	85800203
.			.			市委办公二处	85800105
.			.			.	
旅游	32	12	综合类大学	325	325	华中科技大学	87870001
			理工大学	430		武汉大学	86880206

图 1.1.1 电话号码簿

大、小类别可以看做电话号码簿的目录或索引。如果电话号码簿缺少类别索引,而是按照电话安装先后的顺序进行排列或毫无规律地排列,用户会使用这样的电话号码簿吗?如果电话号码簿只记载着一个部门的十几部或几十部电话,还需要分类吗?

问题就是数据量的多少。少量数据的查找无需考虑数据的组织形式,而大量数据如果没有合理的组织形式,查找过程就只能顺序地进行,从而导致用户无法接受;在合理的数据组织结构下,用户会按照从大类别到小类别,然后在确定的小范围内顺序检索到所需要的信息。我们将检索过程称为检索的算法,每次检索都是按照同一规律进行的;电话号码簿的变更不会改变其数据的分类结构和数据的检索算法。

可见,这个问题的数据模型是对数据的两级分类索引结构,相应检索算法是先大类,后小类,然后在小类中顺序查找。无论是检索还是变更,都要保证数据模型本身的完整性。

类似地,当用户在一本厚厚的汉语字典查找某一个汉字时,首先必须知道所使用的字典的编码方法,然后才能按照偏旁部首、或者拼音等相应的编码方法较快地查到所需查找的汉字。用户能如此顺利地在几万个汉字中找到所需要的汉字,是因为字典中的每一个汉字都是按偏旁部首或者拼音的规律严格地安排在它应处的页行(编码)上。倘若不按某一规律,将几万个汉字任意安排,用户为了查找某一个汉字就不得不从字典的第一页开始逐页查找了。

可以看出查找效率是与字典中汉字的安排规律,也就是数据组织形式密切相关的。自然,对于不同组织形式必须采用相应的查找办法才能达到提高效率的目的。

问题二:车皮调度问题。

有若干个发往同一方向不同城市的货车车皮以随机的次序到达货站的进车道,如图 1.1.2 所示,整个货站由三部分构成:进车道、出车道和调度道(中间多条车道)。

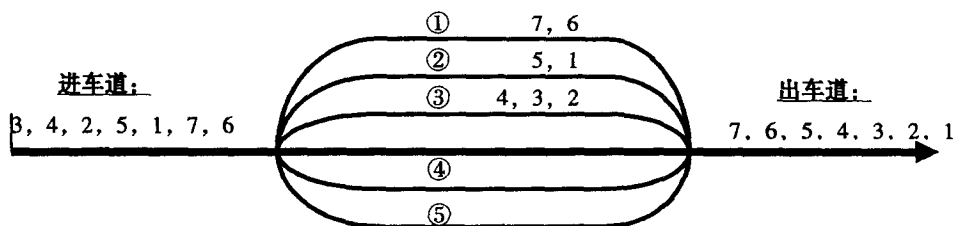


图 1.1.2 车皮调度问题

由于货车的车皮从当前货站发往不同目的地,所以,货车发出本站时,列车的尾部所挂接的车皮应该是发往距本货站最近的车皮,这样可以保证到达某车站时从尾部“甩下”若干到站的车皮。

可问题是,不同的货运委托用户的车皮是以随机的次序到达货站的进车道,为了便于说明,我们假设发往较远站点的车皮的编号比发往较近站点的车皮的编号要小,从图中可以看到进车道上是不同的货运委托用户的车皮到达的次序。这些车皮发出本站前应调整它们的次序,调整为小编号在前、大编号在后的排列次序,即出车道上的次序。

为实现调整过程,货运站需要借助调度车道完成,调度车道由若干条备用车道组成,调度过程可描述为:

将备用车道上第一个到达的车皮(编号为 6 的车皮)任选一备用车道进入;

后面到达的车皮,如果其编号大于它前一个车皮的编号,则进入前一个车皮所进入的备用车道(编号为 7 的车皮, $7 > 6$, 则进入备用道①);

如果其编号小于它前面一个车皮的编号,则另选一备用车道(备用道②)进入(编号为 1 的车皮, $1 < 7$, 则进入备用道②);

如此下去,所有车皮全部进入备用车道,可以看到备用车道上的车皮是有序地排列。

然后将备用车道上的所有车皮进入出车道,挂成一列车,挂接算法是:逐个地从备用车道中选出编号最小的车皮进入出车道,直至备用车道上的车皮全部进入出车道。

这一问题是数值计算无法完成的,它属于非数值计算问题。数据在调度过程中构成了有序“队列”,对于每个车皮的进入,采用同一运算方法进入备用车道。这就是数据结构中讨论的“队列”结构问题,从一头进入,从另一头出去。

问题三:某省各城市之间要架设电话通信线路,既要保证各城市间互通,又要使架设成本最少,就是数据结构中讨论的图结构的应用——最小生成树。

图 1.1.3 给出了各城市之间的距离,要实现各城市间互通,就要城市之间都有线路连接;要使成本最少,就要选择较近的城市之间架设线路,因此应该采用的架设结构如图 1.1.4 所示。

数据的组织结构称为“图”结构,算法过程称为求取“最小生成树”。

可以看到算法过程就是保留连接线路中值最小的连接线路,且保证任何一城市都至少与另一个城市相连通。这样既保证了城市之间“连通”,又使“连通”成本最小。

这个问题中数据的组成结构是由顶点(城市)集合和边(线路)集合构成,求得的结果中,顶点(城市)集合没有发生改变,边(线路)集合减少了,但整个数据结构仍然是“图”结构。

运算过程（以后再讨论）是一定的。

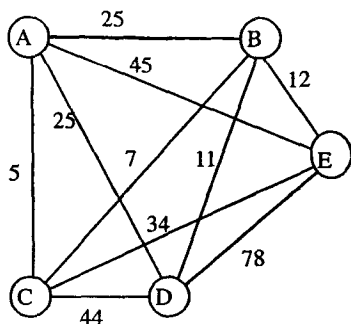


图 1.1.3 城市连接及距离

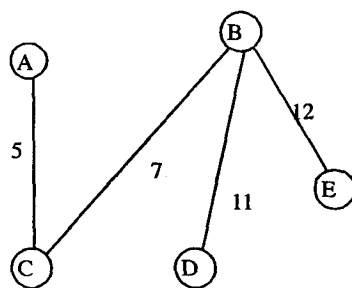


图 1.1.4 最小生成树

对上述三个问题进行归纳可以发现：

第一个问题中，数据可以看做由三个表存放，表与表之间的关系是由每个表项中的页码关联的；每个表中的数据之间的关系是由前后存放的次序进行关联的；检索时，前两个表之间利用页码指向跳跃查找，在第三张表中则顺序查找。这个问题涉及大量数据，所以在处理中要合理排列数据。

第二个问题中，数据在进车道是按照先来排在前、后来排在后、不允许中间插入的原则形成一个队列；在调度过程中，将进车道中的队列分散到不同的备用车道，每个备用车道又是一个有序的子队列；最后，再形成一个有序的完整队列进入出车道。每个队列中元素以进入的先后进行关联。这个问题的处理中，总是从队列的一端进入数据，从另一端移出数据。

第三个问题中，数据是由两个集合组成，“顶点”集合中存放着顶点名称，“边”集合中存放着边的长度和边所关联的是哪两个顶点，两个集合中的数据可以是无序的；运算过程是找出较小的边，且保证保留的边使所有顶点间“连通”，但不形成环形路。这个问题的数据存放为两个集合。

从上面的三个问题可以看出，这些问题的模型都无法用数学的公式或方程或函数来解决，它们都是“非数值计算的问题”。

1.1.2 数据结构的定义

综上所述，我们可以这样给数据结构这门学科下个定义：数据结构就是研究计算机中大量数据存储的组织形式，并定义且实现对数据的相应的运算，以提高计算机的数据处理能力的一门科学。

对该定义需要说明的是，数据的组织形式（结构）具有两个层面：

一个是与计算机本身无关的“逻辑组织结构”，简称“逻辑结构”，它的构成是由数据的值、数据与数据之间的关联方式两个部分组成。如电话号码簿中【大类名、大类所指的页码】、【小类名、小类页码】、【单位名、单位的电话号码】是数据的值；数据间的关联是由大类所指的页码和小类页码共同实现的。

另一个是如何将具有逻辑组织结构的数据在计算机的存储介质上存放的“物理组织结构”，简称“物理结构”。如电话号码簿中，每一张表内的数据可以在存储介质上连续地顺序存放，但表与表之间是在非连续的存储介质上分开存放的。

“运算”或“操作”是数据结构讨论内容的一个核心问题。不同实际问题具有不同的处理要求,所有的处理要求要进行事先定义,并用计算机的某种语言给予描述,这就是“算法设计”。

算法不仅要实现问题的要求,而且应该是高效地完成。低效的算法无法满足用户的需求或根本不能运用于实际。用一个高效的处理算法设计的程序结合高速运算的计算机可以最大程度地满足用户处理要求,而一个低效的处理算法设计的程序即使运用高速运算的计算机也不能满足用户的处理要求。

对实际问题的处理能力不仅取决于计算机硬件本身的处理能力,而且更多地取决于对数据结构的合理组织以及相应处理算法的优劣,

1.2 数据结构的相关概念

为了更好地描述数据结构的内容,我们将用到许多术语。对相关的术语,我们在本节中给出确切的含义,以便在今后的学习中能有统一的概念。

1.2.1 数据和信息

在计算机中,数据这个名词的含义异常广泛,可以认为它是描述客观事物的数字、字符,以及所有能输入到计算机中并能为计算机所接受的符号集合的总称。

数据是一个抽象的概念,它是一组符号的集合。单个符号是无法描述现实的,只有将多种符号进行组合,才可以描述现实。如单个字母或字符无法作为语言,字母或字符的组合才能形成语言。

数据是信息以某一类特定符号表示的形式,是计算机程序加工的对象。随着 IT 技术的发展,数据所能描述的信息越来越丰富,多媒体信息中的视频信号、音频信号经过专门设备的采集和转换后成为数字符号,形成计算机可存储并操作的数据。

1.2.2 数据元素

数据元素是数据集中的个体,是数据组成的基本单位。数据通常是由若干个数据元素组成,数据元素是不可再分的最小单位。在数据存储组织中,它是基本的处理单位。

如在商品信息的描述中,每个商品数据就是一个数据元素,商品数据元素的构成是:【商品名、商品编号、商品价格、商品数量】,其中“商品价格”又可以再分为:【出厂价格、批发价格、零售价格】。如图 1.2.1 所示。

构成商品数据元素的每个项目称为“数据项”(data item),有的数据项是由单一的数据类型组成,称为原子项,而有的数据项又可再分为若干个子数据项组成,称为组合项,“商品价格”就是一个组合项。

数据元素是数据结构理论中较为抽象的概念,在具体描述中通常又表达为“记录”、“结点”或“顶点”。在计算机的高级语言中又可以有不同的表述,如 C 语言中定义为“结构类型”(structure type),而在 Pascal 语言中定义为“记录类型”(record type)。在数据库理论中定义为“元组”(tuple)。

在数据元素的多个数据项中能起标识作用的数据项称为关键字或关键码(key)。所谓标识作用即确定或识别数据元素的作用,如商品数据元素中,【商品编号】是惟一能标识某一商品数据的数据项,它就是商品数据元素的关键字。关键码如能惟一标识一数据元素,又称为主

关键码,反之称为次关键码或次码。

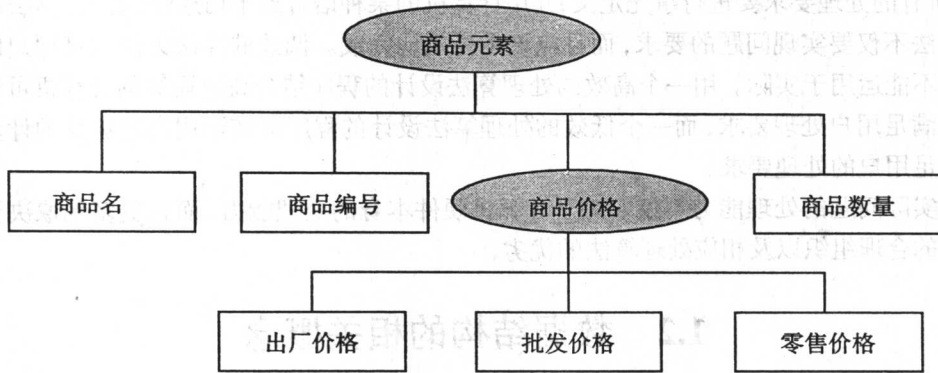


图 1.2.1 商品数据元素

1.2.3 结构类型

数据结构中讨论的结构类型分为两个层面,一是逻辑层面的数据结构,简称逻辑结构;另一个是物理层面的数据结构,简称物理结构。

1. 逻辑结构

逻辑结构描述数据元素与数据元素之间的关联方式,简称为关系,表示的是事物本身的内在联系。逻辑结构又可以分为:线性结构和非线性结构两大类。

线性结构的特点表现为,数据元素之间存在前后次序,排在某个数据元素 b 前面的数据元素 a 称为 b 的直接前驱元素,而数据元素 b 则称为数据元素 a 的直接后继元素。对于某个数据元素,如果存在直接前驱元素或直接后继元素,则都是惟一的(如,队列结构)。线性结构中数据元素之间的正逆关系都是“一对一”的。线性结构又可再分为:线性表、堆栈、队列等。

非线性结构的特点表现为:数据元素不一定存在确定的前后次序,甚至是无序的,数据元素之间存在从属或互为从属的关系或离散关系。非线性结构又可再分为树形结构、图状或网状结构、纯集合结构。

在树型结构中,数据元素之间存在着“一对多”的关系。某个数据元素 a (结点)可能有多个分枝,每个分枝所联接的数据元素从属于数据元素 a (多个数据元素从属于一个数据元素)。

在图或网状结构中,数据元素之间存在着“多对多”的关系。如城市间的公路问题,每个城市数据元素与多个城市数据元素相邻接,这些邻接数据元素相互之间不存在从属关系或理解为相互从属。

在纯集合结构中,数据元素具有“同属于一个集合”的关系。

纯集合结构和集合结构是不同的,纯集合结构中只有数据元素本身,而不存在数据元素之间的“联接”关系;集合结构中除了有数据元素本身组成的子集外,还存在数据元素之间的“联接”关系所组成的关系子集合。

所以说,所有逻辑结构类型都可以看做集合结构类型或集合结构类型的特例。例如,在集合结构中去掉一些“关系”,使每个数据元素联接为“一对一”,这时,集合结构就变成成为线性结构。

2. 物理结构

物理结构也称为存储结构,是逻辑结构的数据元素在计算机的物理存储空间上地映象,映象不仅包含数据元素本身,而且包含着数据元素之间的关联方式,即关系的映象。

映象表现为两种方式:顺序映象和非顺序映象。

(1) 顺序映象。顺序映象是指数据元素连续地在一块物理存储空间上存储,物理存储空间只用于存放数据元素本身,数据元素之间的关联以两个数据元素存储的相邻关系来表示或通过某个函数来表示。或者说,利用数据元素在存储空间上的相对位置来表示数据元素之间的逻辑关系。

如一组成绩信息,每个数据由姓名和成绩两个数据项构成:

{ { 彭亮, 97 }, { 王明, 95 }, { 李智, 90 }, { 刘丹, 88 }, { 肖象, 78 } }

数据元素是按成绩从高至低的顺序排列,即按成绩从高至低关联,在物理存储空间上的存储映象如图 1.2.2 所示。

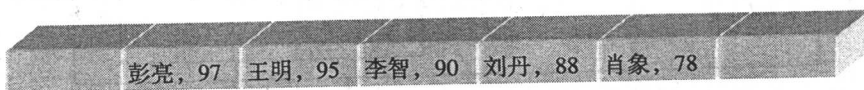
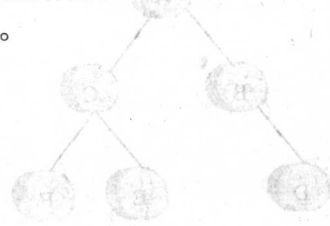


图 1.2.2 成绩信息的顺序映象

数据元素存储的空间是连续的,每个数据元素以相邻方式存储,相邻的两个数据元素左边元素的成绩大于右边的元素成绩。数据元素的前驱和后继关系是以数据元素存储的空间相邻性表示的。如要查找第 i 名的成绩,则从第 i 个空间中可以得到。

再如,有一个下三角矩阵:

	1	2	3	4	5
1	0	0	0	0	0
2	A	0	0	0	0
3	B	C	0	0	0
4	D	E	F	0	0
5	G	H	I	J	0



在物理存储空间上存放为如图 1.2.3 所示,数据元素存储的空间是连续的,每个数据元素以相邻方式存储,数组元素按行优先法则存储。如要查找第 i 行、第 j 列 ($1 \leq i \leq 5, 1 \leq j \leq 5$) 数据元素,则从第 $((i-1) \times (i-2))/2 + j$ 个数据元素空间中可以得到,即数据元素关系函数为:

$$F(i,j) = ((i-1) \times (i-2))/2 + j$$

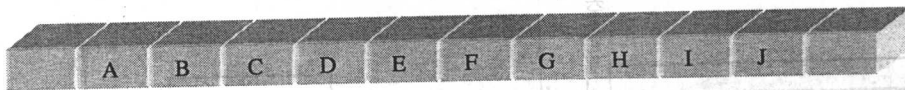


图 1.2.3 下三角矩阵的顺序映象

从上面两个问题中可以看出,数据元素顺序地映象在连续的存储的空间上,特别是矩阵从逻辑上看是有行列之分,但在物理存储空间上是无行列之分,逻辑上的行列关系是通过 F 函数来表示的, F 函数的值就是存储介质上某个空间的地址。