

数据结构

(C语言版)

主 编 吴玉蓉 李 劲



中国水利水电出版社
www.waterpub.com.cn

数据结构

(C语言版)

主 编 吴玉蓉 李 劲
副主编 李 帆 丁益祥 文小华



ISBN 7-111-14805-3
定价：39.00元

中国水利水电出版社
北京市复兴路甲23号
181mm×260mm 16开本 19.5印张 422千字
2008年12月第1版 2008年12月第1次印刷
5001-4000册

北京水利科学研究所
北京水利科学研究所
北京水利科学研究所
北京水利科学研究所



中国水利水电出版社
www.waterpub.com.cn

内 容 提 要

本书为普通高等教育教材,着重介绍数据结构及处理技术。其主要内容有线性数据结构的线性表、栈、队列以及串;非线性数据结构的数组、广义表、树、二叉树、图;数据处理技术的查找、排序、文件的存储结构和组织等。各章配有丰富的典型例题和各种难度的习题及参考答案。对于典型的数据结构和常用的数据处理技术还配有上机实验指导。

为适应广泛应用,本书采用C语言描述所有的算法,阐述本着深入浅出、简洁明了的原则;内容组织注重理论联系实际,教学和学辅融为一体。为便于教学和自学,本书配有电子教案、电子版的源程序以及电子版的习题解答等。

本书可作为高等院校计算机类、信息类、电子类、自动化类以及教育技术类各专业的本科学生、高职高专学生的数据结构课程教材,也适用于从事计算机工程与应用的科技工作者参考,并对计算机等级考试亦具有参考价值。

图书在版编目(CIP)数据

数据结构: C语言版 / 吴玉蓉, 李劲主编. —北京: 中国水利水电出版社, 2008

ISBN 978-7-5084-5781-9

I. 数… II. ①吴…②李… III. ①数据结构—高等学校—教材②C语言—程序设计—高等学校—教材 IV. TP311.12
TP312

中国版本图书馆CIP数据核字(2008)第113127号

书 名	数据结构(C语言版)
作 者	主编 吴玉蓉 李劲
出版发行	中国水利水电出版社(北京市三里河路6号 100044) 网址: www.waterpub.com.cn E-mail: sales@waterpub.com.cn
经 售	电话: (010) 63202266(总机)、68367658(营销中心) 北京科水图书销售中心(零售) 电话: (010) 88383994、63202643 全国各地新华书店和相关出版物销售网点
排 版	中国水利水电出版社微机排版中心
印 刷	北京市兴怀印刷厂
规 格	184mm×260mm 16开本 19.5印张 462千字
版 次	2008年12月第1版 2008年12月第1次印刷
印 数	0001—4000册
定 价	39.00元

凡购买我社图书,如有缺页、倒页、脱页的,本社营销中心负责调换

版权所有·侵权必究

前 言

随着计算机技术的快速发展,计算机软件设计和开发已经渗透到社会的各个领域。数据结构不再仅是计算机专业的必修课程,而且也是非计算机专业如电子类、信息类等专业的必修课程或者选修课程。为满足社会的需求和教学的需要,作者在多年教学经验的基础上,根据学生的认识规律,循序渐进地讲述了数据结构的有关概念、算法等。本书适用于不同专业的需求。

本书共分8章,第1章主要讲述数据结构的基本概念以及算法的基本知识;第2章至第5章主要讲述几种基本的数据结构,即线性表(包括线性表、栈、队列、串)、数组和广义表、树和二叉树以及图;第6章和第7章分别讲述数据处理中常使用的查找技术和排序技术;第8章简单讲述了文件的存储结构和组织方法。

本书编写特点如下:

1. 通俗易懂。基础理论知识的阐述由浅入深、通俗易懂,内容引出强调逻辑衔接,遵循人们的认识习惯。在内容的组织和编排上略去了一些理论上的推导和繁琐的数学证明,在典型的算法介绍中深入浅出、简洁明了。

2. C语言描述。充分利用发展较为成熟的C编译器和C程序开发环境,将所有的算法均采用“C语言描述”,这是因为不仅大多数读者此前都已学习过C语言,而且C语言已经是个人计算机和UNIX工作站的主要开发语言,计算机科学的编程领域中许多概念如虚拟内存、文件系统、解析器的自动生成、词法分析器以及网络等,都是用C实现的。

3. 理论与实践相结合。突出能力训练,相对传统计算机专业的《数据结构》教材而言,加大了基础题和应用题的训练,基础题是在各级各类考题中精选出来的,应用题是作者们在教学过程中、在工程实践中积累起来的典型实例。通过这些题目的练习,不仅能加深对各种数据结构的基本概念和算法的理解,而且通过上机实验,能够提高程序设计的能力。

4. 教材和学辅融为一体。教材的每一章学习内容后面均有本章小结并配有大量的基础题、应用题以及上机实验题。书后附录中给出了参考答案,使得读者在拥有一本教材的同时又拥有了一本学习辅导书。

本书第1章和第3章由李劲编写；第2章和第4章由文小华、向静和高发桂编写；第5章由李帆编写；第6章由丁益祥编写；第7章由吴玉蓉编写；第8章由张国琴编写。全书由吴玉蓉统稿。

在本书的编写过程中，编者参考了大量的有关数据结构的书籍和资料，在此对这些参考文献的作者诚表衷心的感谢。

由于编者水平有限，书中的错误和不足之处在所难免，恳请读者批评指正。

作者

2008年11月

目 录

前言

第 1 章 数据结构的基本概念及算法

- 1.1 概述 1
- 1.2 基本概念及术语 3
- 1.3 算法 7
 - 1.3.1 算法的定义及特点 8
 - 1.3.2 算法的描述 8
 - 1.3.3 算法设计的要求 9
 - 1.3.4 算法效率的分析 9
 - 1.3.5 算法存储空间的分析 11
- 小结 11
- 习题 1 12

第 2 章 线性表

- 2.1 线性表的定义及基本运算 14
 - 2.1.1 线性表的定义 14
 - 2.1.2 线性表的基本运算 14
- 2.2 顺序存储结构的线性表 15
 - 2.2.1 线性表的顺序存储结构 15
 - 2.2.2 顺序表的基本运算 16
 - 2.2.3 顺序表算法的复杂度分析 18
- 2.3 链式存储结构的线性表 18
 - 2.3.1 单链表 19
 - 2.3.2 循环链表 25
 - 2.3.3 双向链表 26
- 2.4 特殊的线性表之一——栈 28
 - 2.4.1 基本概念 28
 - 2.4.2 顺序栈 29
 - 2.4.3 链栈 32
- 2.5 特殊的线性表之二——队列 34
 - 2.5.1 基本概念 34
 - 2.5.2 顺序队列 35

2.5.3	循环队列	36
2.5.4	链式队列	39
2.6	特殊的线性表之三——串	41
2.6.1	基本概念	41
2.6.2	串的存储结构	42
2.6.3	串的基本运算	44
2.6.4	串的模式匹配	46
	小结	54
	习题 2	55
	上机实验 1 线性表的基本操作	60
第 3 章	数组及广义表	62
3.1	数组	62
3.1.1	定义	62
3.1.2	基本操作	63
3.1.3	数组的顺序存储结构	63
3.1.4	规则矩阵的压缩存储	65
3.2	稀疏矩阵	67
3.2.1	稀疏矩阵的三元组表示	67
3.2.2	稀疏矩阵的十字链表表示	72
3.3	广义表	76
3.3.1	定义	76
3.3.2	广义表的存储结构	77
3.3.3	广义表的基本操作	77
	小结	78
	习题 3	79
	上机实验 2 数组的基本操作	81
第 4 章	树和二叉树	82
4.1	树的基本概念与存储表示	82
4.2	二叉树	84
4.2.1	二叉树的基本概念	84
4.2.2	二叉树的基本性质	85
4.2.3	二叉树的存储结构	86
4.3	二叉树的遍历	88
4.3.1	先序遍历	88
4.3.2	中序遍历	89
4.3.3	后序遍历	89
4.3.4	二叉树的建立	90

4.3.5	二叉树遍历的应用	91
4.4	线索二叉树	93
4.5	树、森林与二叉树的转换及遍历	95
4.5.1	树的二叉树表示	95
4.5.2	森林与二叉树的转换	97
4.5.3	树、森林的遍历	97
4.6	哈夫曼树及其应用	97
4.6.1	哈夫曼树的基本概念	98
4.6.2	哈夫曼树的构造	99
4.6.3	哈夫曼树在编码中的应用	100
	小结	103
	习题 4	103
	上机实验 3 二叉树的基本操作	106
第 5 章	图	108
5.1	基本概念	108
5.1.1	图的定义	108
5.1.2	图的相关术语及基本概念	108
5.1.3	图的基本运算	111
5.2	图的存储结构	111
5.2.1	邻接矩阵	111
5.2.2	邻接表	113
5.2.3	邻接多重表	116
5.2.4	十字链表	117
5.3	图的遍历	119
5.3.1	深度优先搜索遍历	119
5.3.2	广度优先搜索遍历	121
5.4	最小生成树	122
5.4.1	生成树和最小生成树	122
5.4.2	普里姆算法	123
5.4.3	克鲁斯卡尔算法	126
5.5	最短路径	128
5.5.1	单源最短路径	128
5.5.2	每对顶点之间的最短路径	132
5.6	拓扑排序	133
5.6.1	AOV 网和拓扑排序的概念	134
5.6.2	拓扑排序的算法	135
	小结	137

习题 5	137
上机实验 4 图的基本操作	141
第 6 章 查找	143
6.1 基本概念	143
6.2 线性查找	144
6.2.1 顺序查找	144
6.2.2 对分查找	146
6.2.3 分块查找	149
6.3 树表查找	150
6.3.1 二叉排序树查找	151
6.3.2 多层索引树查找	156
6.4 哈希查找	162
6.4.1 哈希表与哈希函数	162
6.4.2 构造哈希函数的常见方法	164
6.4.3 冲突解决方法	166
6.4.4 哈希表的查找及分析	167
小结	169
习题 6	169
上机实验 5 查找的基本操作	171
第 7 章 排序	173
7.1 基本概念	173
7.2 内排序	175
7.2.1 交换类排序	175
7.2.2 插入类排序	181
7.2.3 选择类排序	185
7.2.4 其他排序方法	192
7.2.5 内排序方法比较	198
7.3 外部排序简介	199
小结	200
习题 7	200
上机实验 6 排序的基本操作	203
第 8 章 文件	205
8.1 基本概念	205
8.1.1 文件的逻辑结构和物理结构	206
8.1.2 文件的操作(运算)	206
8.2 顺序文件	206
8.3 索引文件	207

8.4 索引顺序文件	208
8.4.1 ISAM 文件	208
8.4.2 VSAM 文件	210
8.5 散列文件	212
8.6 多关键字文件	212
8.6.1 多重表文件	212
8.6.2 倒排序文件	214
小结	215
习题 8	215
模拟试题一	216
模拟试题二	219
附录一 习题参考答案	223
习题 1	223
习题 2	224
习题 3	232
习题 4	235
习题 5	243
习题 6	252
习题 7	256
习题 8	263
附录二 上机实验参考程序	265
上机实验 1	265
上机实验 2	274
上机实验 3	276
上机实验 4	280
上机实验 5	283
上机实验 6	291
附录三 模拟试题参考答案	297
模拟试题一	297
模拟试题二	298
参考文献	301

第1章 数据结构的基本概念及算法

1.1 概 述

随着计算机技术的发展,计算机的应用领域不断扩大,其中利用计算机进行数据处理就是计算机应用的一个重要领域。在进行数据处理时,需要处理的数据元素通常会很多,而这些大量的数据元素都需要存放在计算机中。因此,大量的数据元素按什么结构存放在计算机中,可以提高数据处理的效率,节省计算机的存储空间,已成为进行数据处理的关键问题。

数据结构作为一门学科,主要研究和讨论数据元素之间的逻辑结构、数据元素的存储结构以及对数据结构进行的运算这三方面的问题,其目的是为了提提高数据处理的效率。提高数据处理的效率主要包括两个方面:一是提高数据处理的速速度;二是尽量节省在数据处理过程中所占用的计算机存储空间。

为了增加对数据结构的感性认识,下面举例说明数据结构的概念。

【例 1.1】 无序表的顺序查找与有序表的对分查找。

假设某校选修课成绩登记表和学生情况登记表见表 1.1 和表 1.2。在表 1.1 中,学生的排列顺序是没有规律的,即为无序表。在表 1.2 中,每个学生是按学号从小到大顺序排列的,故为有序表。

表 1.1 2007~2008 学年第一学期 matlab 语言选修课成绩登记表

学 号	姓 名	性 别	专 业	成 绩
0503101152	余雨晴	女	电子科学技术	优
0503101078	黄宇虹	男	电子科学技术	良
0503101001	吴友	男	自动化	中
0503101056	王靖	男	自动化	良
0503101401	涂玉芹	女	电子信息工程	中
0503101434	黄厚锐	男	电子信息工程	及格

表 1.2 2006 级自动化专业学生情况登记表

学 号	姓 名	性 别	出生年月	家庭地址
0603081001	宋立超	女	1987.6	湖北荆州
0603081002	莫甜	女	1987.3	湖北恩施
0603081003	江万春	女	1987.12	湖南湘潭
0603081004	姜天彪	男	1986.2	河南郑州
0603081005	马明哲	男	1986.6	广西柳州
0603081006	张小成	男	1987.5	广东东莞
0603081007	余鹏	男	1986.9	湖北武汉

下面考虑在这两个表中进行查找的问题。

(1) 在表 1.1 中的无序表中进行查找。表 1.1 中,若要查找某位学生的记录,必须从表的第一个记录开始,逐个将表中的记录与所给学生记录进行比较。若表中没有找到所给学生记录,则查找失败;若表中某个记录与所给学生记录完全相同,则查找成功。这种查找方法称为顺序查找。在顺序查找中,如果被查找的记录在表的前部,则需要比较的次数就少;如果被查找的记录在表的后部,则需要比较的次数就多。请思考如果被查找的记录分别是表中第一个、最后一个或者表中根本就不存在这个记录,则需要与表中的记录比较多少次?

当表很大时,顺序查找是很费时间的。如果待查找的表是无序表,即使顺序查找很费时间,也没有更好的查找方法,只能顺序查找。

(2) 在表 1.2 所示的有序表中进行查找。有序表中学生的记录是按学号从小到大的顺序排列的,如果采用有序表的对分查找方法,可以提高查找的效率。

有序表的对分查找是将被查找元数与表的中间元素进行比较;若相等,则表示查找成功,结束查找;若被查找数大于表的中间元素,则表示被查找数在表的后半部分,此时可以抛弃表的前半部分而保留表的后半部分;若被查找数小于表的中间元素,则表示被查找元素在表的前半部分,此时可以抛弃表的后半部分而保留表的前半部分。然后对剩下的部分再按上述方法进行查找。这个过程一直做到在某一次的比较相等(查找成功)或剩下部分已空(查找失败)为止。这种查找方法称为有序表的对分查找。

对分查找只适合于有序表,对于无序表是无法进行对分查找的。

【例 1.2】 学生信息查询系统。

如果需要查询某个专业或某个年级的学生信息时,只要建立相关的学生信息数据结构,按照某种算法编写相关程序,就可以实现计算机自动查询。

由此,可以在学生信息查询系统中建立一张以学号顺序排列的学生信息表(见表 1.3)和姓名索引表(见表 1.4)、专业索引表(见表 1.5)、年级索引表(见表 1.6),表 1.3~表 1.6 构成的文件就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。

表 1.3

学生信息表

学 号	姓 名	性 别	专 业	年 级
070002	艾志民	男	电科	2007 级
061004	刘丽	女	电信	2006 级
050503	王刚	男	自动化	2005 级
040318	古江波	男	自动化	2004 级
040527	戴宝国	男	电信	2004 级
050526	张凯	男	电科	2005 级
061034	宋立超	女	自动化	2007 级

表 1.4 姓名索引表

艾志民	070002
戴宝国	040527
古江波	040318
刘丽	061004
宋立超	061034
王刚	050503
张凯	050526

表 1.5 专业索引表

自动化	040318
	050503
	061034
电信	040527
	061004
	050526
电科	070002

表 1.6 年级索引表

2004 级	040318
	040527
	050503
2005 级	050526
	061034
2006 级	061004
2007 级	070002

通过上述两个例子可知,数据在表中的排列顺序和数据的组织方式,都会影响查找的效率。

在计算机上实现学生成绩的管理,必然涉及到 3 个问题:如何组织学生成绩表?采用何种存储方式将表中的数据及数据之间的关系存放到计算机的存储器中?在计算机上如何完成学生成绩系统的管理功能?例如新生报到后,要添加新学生的记录;当老生毕业时,要删除该生的相关记录;当某个学生转专业了,要修改记录等等。所以数据结构必须给出相应的插入、删除、修改等运算。这些就是数据结构所要讨论的问题。

综上所述,数据结构可以定义为选择适当的组织方式按照某种关系来组织大量的数据,以一定的存储方式把它们存储到计算机中,并在这些数据上定义一个相应的运算,以提高计算机的数据处理能力的一门学科。

1.2 基本概念及术语

在计算机科学中,数据是计算机程序加工处理的对象,数据是对客观事物所进行的描述,而这种描述是采用了计算机系统所能识别、存储和处理的形式来进行的。总之,数据(data)在计算机科学中是指所有能输入到计算机中并被计算机程序处理的符号的总称。因此,对计算机而言,数据的含义很广,可以分为数值型数据和非数值型数据,前者包括整数、实数等,后者包括文字、图形、声音、图像等。

数据元素(data element)是数据的基本单位,在计算机程序中通常作为一个整体进行考虑和处理。在不同的情形下,数据元素又可称为元素、结点、顶点、记录等。有时,一个数据元素可由若干个数据项组成。数据项(data item)是具有独立含义的数据最小单位,也称为字段或域。能够唯一标识一个数据元素的数据项称为关键字(key)。

数据对象(data object)是性质相同的数据元素的集合,是数据的一个子集。例如:某班级本学期《数据结构》期末成绩集合{23, 89, 98, ...}。

数据结构(data structure)是指数据以及相互之间存在一种或多种特定关系的数据元素的集合。数据结构主要研究和讨论以下三个方面的问题:①数据元素之间所固有的逻辑关系,即数据的逻辑结构;②数据元素及其关系在计算机中的存储方式,即数据的存储结构,也称为数据的物理结构;③施加在数据上的操作,即数据的运算。

(1) 数据的逻辑结构 (logical structure)。数据的逻辑结构是指反映数据元素之间逻辑关系的数据结构, 包括两个要素: 一是数据元素的集合, 通常记为 D ; 二是数据元素间的关系, 通常记为 R 。为了更确切地描述一种数据结构 (记为 B), 通常采用二元组表示:

$$B = (D, R)$$

若 D 为空集, 那么 B 也就无结构可言, 有时把这种情况认为是具有任一结构。若 D 非空, 而 R 为空, 则表明集合 D 中的结点间不存在任何关系, 彼此是独立的。

D 上的一个关系 R 是序偶的集合, 对于 R 中的任一序偶 $\langle x, y \rangle (x, y \in D)$ 。称 x 为序偶的第一结点, 称 y 为序偶的第二结点, 又称序偶的第一结点为第二结点的直接前驱 (通常简称前驱), 称第二结点为第一结点的直接后继 (通常简称后继)。如在 $\langle x, y \rangle$ 的序偶中, x 为 y 的前驱, y 为 x 的后继。

若某个结点没有前驱, 则称该结点为开始结点; 若某个结点没有后继, 则称该结点为终端结点。

对于对称序偶, 即 x 是 y 的前驱, y 也是 x 前驱, 则用圆括号代替尖括号 (x, y) 。

根据数据元素间关系的特性可以将数据的逻辑结构分为 4 类:

- 1) 集合。在集合中, 数据元素之间除了“同属于一个集合”的关系之外, 没有其他的任何关系。集合是元素关系极为松散的一种结构。
- 2) 线性结构。此结构中的数据元素存在一对一的关系。
- 3) 树形结构。此结构中的数据元素存在着一对多的关系。
- 4) 图形结构或网状结构。此结构中的数据元素间存在着多对多的关系。

下面分别举例说明。

【例 1.3】 集合型的数据结构。

$B = (D, R)$, 其中 $D = \{25, 67, 12, 45, 98\}, R = \emptyset$ 。

【例 1.4】 线性数据结构。

一年四季的数据结构可以表示为:

$$B = (D, R)$$

$$D = \{\text{春}, \text{夏}, \text{秋}, \text{冬}\}$$

$$R = \{\langle \text{春}, \text{夏} \rangle, \langle \text{夏}, \text{秋} \rangle, \langle \text{秋}, \text{冬} \rangle\}$$

用图形表示该数据结构的逻辑结构。

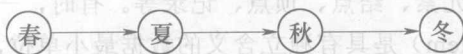


图 1.1 一年四季数据结构的图形表示

解: 对应的图形如图 1.1 所示。

线性结构的特点是开始结点和终端结点都是唯一的, 除了开始结点和终端结点以外, 其他结点有且仅有一个前驱, 有且仅有一个后继。

【例 1.5】 树形数据结构。

书的数据结构可以表示为:

$$B = (D, R)$$

$$D = \{\text{书}, \text{第 1 章}, \text{第 2 章}, \text{第 3 章}, \text{第 4 章}, \text{1.1 节}, \text{1.2 节}, \text{2.1 节}, \text{2.2 节}, \text{2.3 节}, \text{3.1 节}, \text{4.1 节}, \text{4.2 节}\}$$

$R = \{ \langle \text{书, 第1章} \rangle, \langle \text{书, 第2章} \rangle, \langle \text{书, 第3章} \rangle, \langle \text{书, 第4章} \rangle, \langle \text{第1章, 1.1节} \rangle, \langle \text{第2章, 2.1节} \rangle, \langle \text{第2章, 2.2节} \rangle, \langle \text{第2章, 2.3节} \rangle, \langle \text{第3章, 3.1节} \rangle, \langle \text{第3章, 3.2节} \rangle, \langle \text{第4章, 4.1节} \rangle, \langle \text{第4章, 4.2节} \rangle \}$

用图形表示该数据结构的逻辑结构。

解：对应的图形如图 1.2 所示。

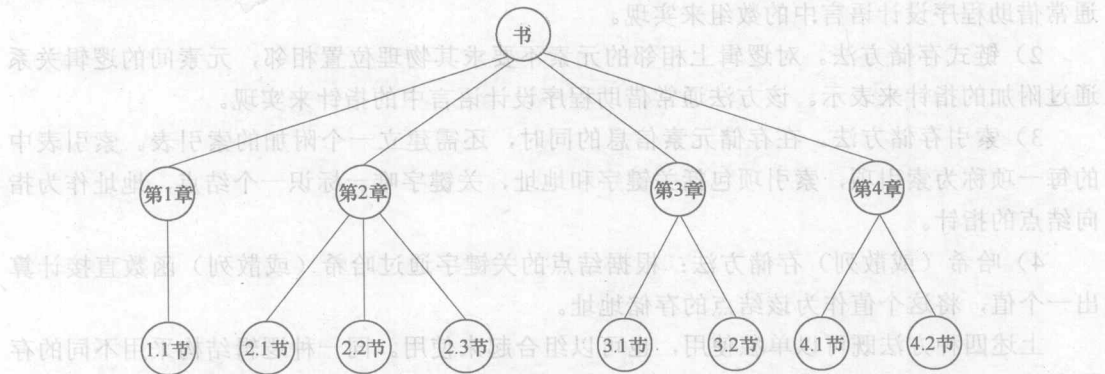


图 1.2 书数据结构的图形表示

树形结构的特点是每个结点最多只有一个前驱，但可以有多个后继，可以有多个终端结点。

【例 1.6】 图形数据结构。

城市交通图的数据结构可以表示为：

$B = (D, R)$

$D = \{ \text{武汉, 北京, 上海, 合肥} \}$

$R = \{ \langle \text{武汉, 北京} \rangle, \langle \text{武汉, 上海} \rangle, \langle \text{武汉, 合肥} \rangle, \langle \text{北京, 上海} \rangle, \langle \text{北京, 合肥} \rangle, \langle \text{上海, 合肥} \rangle \}$

用图形表示该数据结构的逻辑结构。

解：对应的图形如图 1.3 所示。

图形结构的特点是每个结点的前驱和后继的个数可以是任意的。因此，可能没有开始结点和终端结点，也可能有多个开始结点、多个终端结点。

树形结构和图形结构统称为非线性结构，该结构中的结点存在一对多或多对多的关系。由图形结构、树形结构和线性结构的定义可知，线性结构是树形结构的特殊情况，树形结构是图形结构的特殊情况。

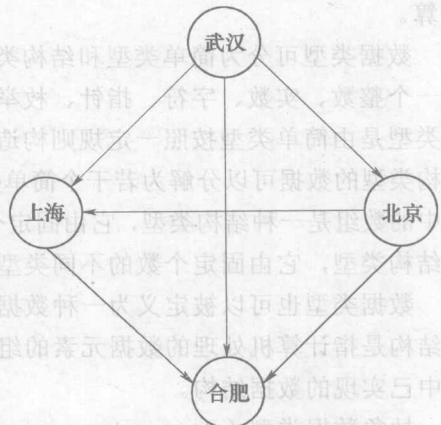


图 1.3 城市交通图数据结构图形表示

逻辑结构从逻辑关系上描述数据的，与数据在计算机中的存储位置无关，是独立于计算机的。而我们研究数据结构的目的是为了在计算机中实现对它的运算，为此，还需要研究如何在计算机中表示一个数据结构，即数据的存储结构。

(2) 数据的存储结构 (physical structure)。数据的存储结构是逻辑结构在计算机中的实现方法 (又称映像), 包括数据结构中元素的表示及元素间关系的表示。常用的数据结构的存储方法有四种:

1) 顺序存储方法。把逻辑上相邻的结点存储在物理位置上相邻的存储单元里, 结点间的逻辑关系由存储单元的邻接关系来体现。顺序存储结构是一种最基本的存储表示方法, 通常借助程序设计语言中的数组来实现。

2) 链式存储方法。对逻辑上相邻的元素不要求其物理位置相邻, 元素间的逻辑关系通过附加的指针来表示。该方法通常借助程序设计语言中的指针来实现。

3) 索引存储方法。在存储元素信息的同时, 还需建立一个附加的索引表。索引表中的每一项称为索引项, 索引项包括关键字和地址, 关键字唯一标识一个结点, 地址作为指向结点的指针。

4) 哈希 (或散列) 存储方法: 根据结点的关键字通过哈希 (或散列) 函数直接计算出一个值, 将这个值作为该结点的存储地址。

上述四种方法既可以单独使用, 也可以组合起来使用。同一种逻辑结构采用不同的存储方法, 可以得到不同的存储结构, 选择哪一种存储方法要根据运算的实际情况来具体确定, 主要考虑的是运算的便利性以及算法的时空要求。

(3) 数据的运算。数据的运算是在数据上施加的一系列操作。数据的运算是定义在逻辑结构上的, 而运算的实现要在存储结构上进行。数据的各种逻辑结构有相应的一组运算。例如, 最常用的运算有查找、插入、删除、更新和排序等。

数据类型 (data type) 是一个值的集合和定义在这个值集上的所有的操作。不同类型的变量, 其所能取值的范围不同, 所能进行的操作也不同。例如, C 语言中的整型变量, 其值集为某个区间上的整数, 定义在其上的操作为加、减、乘、除及取模等算术运算。

数据类型可分为简单类型和结构类型两种。简单类型中的每个数据都是不可分解的, 如一个整数、实数、字符、指针、枚举量等都是无法再分割的整体, 均属于简单类型。结构类型是由简单类型按照一定规则构造而成, 且结构类型中还可以包含结构类型, 故一种结构类型的数据可以分解为若干个简单数据和结构数据, 每个结构数据还可以再分。如 C 语言中的数组是一种结构类型, 它由固定个数的同一类型的数据顺序排列而成; 结构体也是一种结构类型, 它由固定个数的不同类型的数据顺序组合而成。

数据类型也可以被定义为一种数据结构和对该数据结构允许进行的运算集。总之, 数据结构是指计算机处理的数据元素的组织形式和相互关系, 而数据类型是某种程序设计语言中已实现的数据结构。

抽象数据类型 (abstract data type, 缩写为 ADT) 是指一个数学模型以及定义在该模型上的一组操作, 即用户进行软件系统设计时从问题的数学模型中抽象出来的逻辑数学结构和逻辑数学结构上的运算, 而不考虑计算机的具体存储结构和运算的具体实现算法。抽象数据类型和数据类型实质上是一个概念, 它仅取决于数据类型的逻辑性, 而与其在计算机内部如何表示和实现是无关的。

抽象数据类型一般包括数据对象 (即数据元素的集合)、数据关系和基本运算三方面

的内容。抽象数据类型可以用一个三元组 (D,S,P) 来表示。其中 D 是数据对象, S 是 D 上的关系集, P 是对 D 的基本操作集。其基本格式如下:

ADT 抽象数据类型名

```
{  
    数据对象: 数据对象的定义  
    数据关系: 数据关系的定义  
    基本运算: 基本运算的定义  
}ADT 抽象数据类型名
```

其中, 数据对象和数据关系的定义用伪码描述, 数据基本运算的定义格式为

基本运算名 (参数表)

初始条件: 初始条件描述

运算结果: 运算结果描述

基本运算有两种参数: 赋值参数只为运算提供输入值; 引用参数以 & 打头, 除可提供输入值外, 还将返回操作结果, 在 C 语言中, 可以用指针来代替。例如, 抽象数据类型复数的定义为:

ADT complex

```
{  
    数据对象: D={e1,e2,e3 | e1,e2 均为实数}
```

```
    数据关系: R1={<e1,e2>| e1 是复数的实数部分, e2 是复数的虚数部分}
```

```
    基本操作:
```

```
        assigncomplex(&z,v1,v2): 构造复数 Z, 其实部和虚部分别赋予参数 v1 和 v2 的值。
```

```
        destroycomplex (&z): 复数 z 被销毁。
```

```
        getreal (z, &real): 用 real 返回复数 z 的实部值。
```

```
        getimag(z, &imag): 用 imag 返回复数 z 的虚部值。
```

```
        add(z1,z2,&sum): 用 sum 返回两个复数 z1,z2 和的值。
```

```
}ADT complex
```

抽象数据类型有两个重要特征: 数据抽象和数据封装。所谓数据抽象, 是指用 ADT 描述程序处理的实体时, 强调的是其本质的特征、其所能完成的功能以及它和外部用户的接口 (即外界使用它的方法)。所谓数据封装, 是指将实体的外部特性和其内部实现细节分离, 并且对外部用户隐藏其内部实现细节。抽象数据类型需要通过固有数据类型 (高级程序设计语言中已实现的数据类型) 来实现。

1.3 算 法

著名的计算机科学家、PASCAL 语言发明者沃思 (Nikiklaus Wirth) 教授曾提出一个著名的公式:

数据结构+算法=程序

该公式清楚地揭示了算法与数据结构这两个计算机科学重要支柱的重要性与统一性。不能离开数据结构去抽象地分析求解问题的算法, 也不能脱离算法去研究数据结构。通常, 算法的选择常常在很大程度上依赖数据结构。