

可下载教学资料

<http://www.tup.tsinghua.edu.cn>

21世纪普通高校计算机公共课程规划教材

数据结构实用教程

(C语言版) (第二版)

赵波 董靓瑜 主编

李静 郑巍 霍利 编著

梁旭 主审



清华大学出版社

21 世纪普通高校计算机公共课程规划教材

数据结构实用教程

(C语言版) (第二版)

赵波 董靓瑜 主编
李静 郑巍 霍利 编著

清华大学出版社
北 京

内 容 简 介

《数据结构实用教程(C语言描述)》(第二版)是为“数据结构”课程编写的教材。书中首先介绍了数据结构的观念及数据结构研究的逻辑结构、存储结构及运算三方面内容涉及的基本概念;然后针对经典的数据结构,即线性表、栈、队列、多维数组、广义表、树和图的逻辑特征、常用的存储方式及各种基本运算的实现算法做了详细阐述;最后讨论了两种典型运算——排序和查找的各种实现方法。全书采用C语言作为数据结构和算法的描述工具。在附录A中给出了经典结构典型应用的完整C程序,在附录B中给出了习题解析及参考答案。

本书结构清晰、层次分明、深入浅出、通俗易懂,适用面广。可以作为普通高等院校计算机学科和信息类学科本科和专科教材,也可以作为其他理工类专业的选修教材。

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。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

数据结构实用教程: C语言版/赵波等编著.--2版.--北京:清华大学出版社,2012.9

(21世纪普通高校计算机公共课程规划教材)

ISBN 978-7-302-29748-2

I. ①数… II. ①赵… 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. ①TP311.12 ②TP312

中国版本图书馆CIP数据核字(2012)第187534号

责任编辑:高买花 薛 阳

封面设计:何凤霞

责任校对:焦丽丽

责任印制:杨 艳

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦A座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 装 者:北京市密东印刷有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:19.5 字 数:473千字

版 次:2009年9月第1版 2012年9月第2版 印 次:2012年9月第1次印刷

印 数:1~3000

定 价:33.00元

产品编号:048137-01

出版说明

随着我国改革开放的进一步深化,高等教育也得到了快速发展,各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度,通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程(简称‘质量工程’)”,通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建议,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

本系列教材立足于计算机公共课程领域,以公共基础课为主、专业基础课为辅,横向满足高校多层次教学的需要。在规划过程中体现了如下一些基本原则和特点。

(1) 面向多层次、多学科专业,强调计算机在各专业中的应用。教材内容坚持基本理论适度,反映各层次对基本理论和原理的需求,同时加强实践和应用环节。

(2) 反映教学需要,促进教学发展。教材要适应多样化的教学需要,正确把握教学内容和课程体系的改革方向,在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养,为学生知识、能力、素质协调发展创造条件。

(3) 实施精品战略,突出重点,保证质量。规划教材把重点放在公共基础课和专业基础课的教材建设上;特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现教学质量和教学改革成果的教材。

(4) 主张一纲多本,合理配套。基础课和专业基础课教材配套,同一门课程有针对不同层次、面向不同专业的多本具有各自内容特点的教材。处理好教材统一性与多样化,基本教材与辅助教材、教学参考书,文字教材与软件教材的关系,实现教材系列资源配套。

(5) 依靠专家,择优选编。在制订教材规划时要依靠各课程专家在调查研究本课程教

材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主题。书稿完成后要认真实行审稿程序,确保出书质量。

繁荣教材出版事业,提高教材质量的关键是教师。建立一支高水平教材编写梯队才能保证教材的编写质量和建设力度,希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21 世纪普通高校计算机公共课程规划教材编委会

联系人:梁颖 liangying@tup.tsinghua.edu.cn

前言

本书自出版发行以来,受到了广大读者的关注,在此表示衷心的感谢!本着“不断创新、精益求精、锤炼精品”的原则,在广泛收集任课教师和学生的意见和建议的基础上,编写了《数据结构实用教程(C语言版)》第二版。主要修订内容包括以下几个方面。

(1) 对全书各个章节的内容进行了精细的推敲,从内容的组织编排到讲述的条理性和准确性都进行了修改。力求用最简练的方法和最通俗的叙述诠释复杂的软件设计理论和算法,以适应多方面、多层次的众多读者的需求。

(2) 对课后的习题范围和习题类型进行了较大的调整。在帮助学生及时理解和消化所学知识的同时,也为课程结束的考核环节提供了较好的试题范例。

(3) 集中给出了经典结构的应用案例。对提高学生的理论联系实际能力、程序设计能力、综合应用知识的能力会有很大的帮助。同时对完成课程设计教学过程准备了参考资料。

(4) 对各章都新增了小结和本章重点,把所有的内容进行系统化、条理化,归纳总结在一起由本章最重要的概念引出。可以帮助学生掌握所学内容的联系及课程的整个脉络。

(5) 新增了重点习题解析和参考答案。给出了重点和难点问题解题方法和步骤的指导,为学生理解和掌握相关的基础知识、基本概念和基本方法提供了参考。

全书共包含 8 章内容。第 1 章介绍了数据结构的概念、数据结构研究的三方面内容及数据结构的分类;第 2 章和第 3 章介绍了三种最基本的线性结构,即线性表、栈和队列;第 4 章介绍了两种复杂结构,分别是多维数组和广义表;第 5 章和第 6 章介绍了两种经典的非线性结构,即树和图;第 7 章和第 8 章讨论了数据处理过程中使用频率最高的两种典型运算:排序和查找。全书采用 C 语言作为数据结构和算法的描述工具。利用数组、结构体、指针等重要数据类型并结合 C 函数,完成了书中所有基本运算的实现算法。对经典结构的典型运算,书中还给出了简单应用举例的完整 C 程序,旨在帮助学生掌握如何利用数据结构中的基本运算来解决问题。书中所有的算法都经过上机调试并通过。

本书在内容的选取上符合应用型、复合型人才培养目标的要求,组织编排上体现先理论、后应用、理论与应用相结合的原则,注重课程内容的前后联系,理清来龙去脉,强调条理性和系统性,兼顾学科的广度和深度。本书结构清晰、层次分明、深入浅出、通俗易懂,适用面广。可以作为普通高等院校计算机学科和信息类学科本科和专科教材,也可以作为其他理工类专业的选修教材。

本书的主编有着二十多年数据结构课程教学和研究工作的经验,参加编著过多本教材。本书是作者多年教学成果的结晶,在难点内容的叙述及讲解方法上都有独到之处。本次的

修订工作主要由主编完成,其他作者也参与了部分章节的修改工作。编者均为大连交通大学教师。编写过程中参考了大量的著作、教材等资料,在此一并表示感谢。

虽然全体参编人员都尽心尽力、力求完美,但由于水平有限,书中难免出现遗漏或不妥之处,敬请广大读者不吝指正,不胜感谢。

编 者

2012年5月

目 录

第 1 章 绪论	1
1.1 基本术语	1
1.2 数据结构的定义及研究的内容	2
1.2.1 数据的逻辑结构	2
1.2.2 数据的存储结构	5
1.2.3 数据的运算	6
1.3 算法	7
1.3.1 算法的概念及特性	7
1.3.2 算法的描述	8
1.3.3 算法的评价	12
1.4 学习数据结构的意义和目的	16
小结	17
习题	17
第 2 章 线性表	19
2.1 线性表的定义及运算	19
2.1.1 线性表的定义及逻辑特征	19
2.1.2 线性表上运算的定义	20
2.1.3 线性表的存储结构	21
2.2 顺序表	21
2.2.1 顺序表的定义及表示	21
2.2.2 线性表运算在顺序表上的实现	23
2.3 链表	27
2.3.1 链表的定义及形式	27
2.3.2 单链表	28
2.3.3 循环链表	42
2.3.4 双链表	43
*2.3.5 静态链表	45
2.4 顺序表和链表的比较	47
小结	48

习题	49
第3章 栈和队列	53
3.1 栈.....	53
3.1.1 栈的定义及运算	53
3.1.2 顺序栈及运算的实现	54
3.1.3 链栈及运算的实现	56
3.1.4 栈的应用	57
3.1.5 栈与递归	60
3.2 队列.....	63
3.2.1 队列的定义及运算	63
3.2.2 顺序队列及运算的实现	64
3.2.3 链队列及运算的实现	67
3.3 栈与队列的比较.....	69
小结	70
习题	71
第4章 多维数组及广义表	73
4.1 多维数组.....	73
4.2 矩阵的压缩存储.....	75
4.2.1 特殊矩阵	75
4.2.2 稀疏矩阵	78
4.3 广义表.....	83
4.3.1 广义表的定义	83
4.3.2 广义表的运算	85
小结	85
习题	86
第5章 树	88
5.1 树的定义.....	88
5.2 二叉树.....	90
5.2.1 二叉树的定义及性质	90
5.2.2 二叉树上运算的定义	93
5.2.3 二叉树的存储	95
5.2.4 二叉链表上实现二叉树的遍历运算	99
5.3 线索二叉树	102
5.3.1 中序线索二叉链表.....	104
5.3.2 中序线索二叉链表的中序遍历.....	106
5.3.3 利用中序线索实现前序遍历和后序遍历.....	107

5.4	哈夫曼树	108
5.4.1	哈夫曼树的定义及建立	108
5.4.2	哈夫曼编码及译码	113
5.5	树和森林	117
5.5.1	树和森林的遍历定义	117
5.5.2	森林与二叉树的转换	118
*5.5.3	树的存储	119
小结		122
习题		123
第6章	图	126
6.1	图的概念	126
6.2	图的存储	131
6.2.1	邻接矩阵	131
6.2.2	邻接表	134
6.2.3	边集数组	137
6.2.4	图的三种存储方法的比较	138
6.3	图的遍历	139
6.3.1	深度优先搜索遍历	139
6.3.2	广度优先搜索遍历	143
6.3.3	非连通图的遍历	146
6.4	最小生成树	147
6.4.1	普里姆算法	147
6.4.2	克鲁斯卡尔算法	152
6.5	最短路径	154
6.5.1	单源最短路径	155
6.5.2	任意两点间的最短路径	161
6.6	拓扑排序	163
*6.7	关键路径	167
小结		172
习题		173
第7章	排序	176
7.1	排序的基本概念	176
7.2	插入排序	178
7.2.1	直接插入排序	178
7.2.2	希尔排序	180
7.3	交换排序	182
7.3.1	起泡排序	182

7.3.2 快速排序	184
7.4 选择排序	187
7.4.1 直接选择排序	187
7.4.2 堆排序	189
7.5 归并排序	194
*7.6 基数排序	196
7.7 内排序方法的比较	198
小结	200
习题	200
第8章 查找	203
8.1 查找的基本概念	203
8.2 顺序表查找	204
8.2.1 顺序查找	205
8.2.2 二分法查找	205
8.3 索引查找	209
8.3.1 索引表的组织	209
8.3.2 分块查找	211
8.4 树表查找	213
8.4.1 二叉排序树	213
*8.4.2 平衡二叉排序树	220
*8.4.3 B-树	225
8.5 散列表查找	232
8.5.1 散列表的概念	233
8.5.2 散列函数的设计	234
8.5.3 解决冲突的方法	236
8.5.4 散列表的特点	239
小结	240
习题	241
附录A 经典结构的典型应用程序	243
附录B 习题解析及参考答案	270
参考文献	299

第1章

绪论

信息时代的到来,计算机技术得到了飞速发展,计算机的应用领域也随之进一步拓宽,从最初为了复杂的科学计算,发展到现在已经深入人类生活的各个领域。计算机科学是一门研究数据表示和数据处理的科学。目前,计算机处理的对象除了简单的数值数据外,还包括声音、图形、图像、色彩等复杂的多媒体数据,数据的处理量日益庞大。因此,对于结构复杂的海量数据,要开发出一套性能良好的、处理高效的应用软件系统,不仅要根据需要在程序设计的高级语言和软件开发工具,更重要的是要研究数据的不同结构、组织方法及在计算机中的存储表示,在此基础上选择数据处理的不同算法,通过分析和比较选择出针对实际应用最好的解决方案,这正是数据结构这门课程所要解决的问题。

本章主要介绍数据结构课程中用到的基本概念和基本方法,而所有给出的基本概念和基本方法都是围绕数据结构这个最重要的概念演化出来的,其中很多概念和方法会贯穿整个课程始终。

1.1 基本术语

本节给出与数据结构概念相关的一些基本概念和常用术语。

数据(Data)是人们约定的物理符号,可用来表示客观事物及其活动,是信息的载体。在计算机领域中指所有能被计算机识别、存储、处理、传输的符号的集合,是计算机程序加工处理的对象。数据是个广义的概念,除了常用的整数、小数等数值型数据外,还包括文字、声音、图形、图像及色彩等非数值型数据。

数据元素(Data Element)是数据结构中处理数据的基本单位,又可以称为元素、结点、顶点或记录。数据是由数据元素构成的,在程序中数据元素通常作为一个整体进行处理。如何确定数据元素,完全取决于所要处理的对象。例如,把一所学校作为一个处理对象时,学校的各项信息汇总就是一个数据元素,其中包括所有的学生信息;当把一个学生作为一个处理对象时,一个学生的信息就是一个数据元素;而所有的学生信息组成待处理的数据。显然,数据和数据元素是相对而言的。

数据项(Data Item)是构成数据元素不可分割的具有独立含义的最小标识单位。通常数据元素由若干个数据项组成,如学生的信息可由学号、姓名、成绩等三个数据项组成;特殊情况下,数据元素中也可以只有一个数据项。

数据类型(Data Type)是一个值的集合和定义在这个值集上的一组操作的总称。如C语言中整数类型(int),其值集为 $[-32768, 32767]$ 区间的整数,定义在值集上的操作有:加(+)、减(-)、乘(*)、除(/)、取余(%)和自增(++)、自减(--)运算等。现实世界中数据

的形式是多种多样的,不同形式的数据具有不同的属性,在计算机语言中对应不同的数据类型。

计算机高级语言中的数据类型通常分为原子类型和结构类型两类。原子类型的值不可再分,一般都是高级语言提供的基本数据类型,如C语言的整型、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分量按某种结构组成的,是可分解的,它的分量可以是原子的,也可以是有结构的,如C语言中的数组、结构体等。

1.2 数据结构的定义及研究的内容

有了上面的基本概念和常用术语,就可以给出数据结构的概念了。

世界是由客观事物(实体)构成的,而客观事物之间大多数都存在着一定的联系。在数据结构中,用数据来表示客观事物,数据是数据元素的集合,数据元素之间的关系称为结构,因此数据结构可以看成是数据及数据之间的关系,但这只是从数学的角度认识数据结构。在本课程中,重点要讨论用计算机来处理数据,因此只研究数据及数据之间的关系是不够的,还要研究数据如何保存到计算机中并进行相应的处理。

综上所述,可以给出如下的数据结构定义,并根据数据结构定义归纳出数据结构研究的内容。

数据结构(Data Structure)定义:按照某种逻辑关系组织起来的一批数据,用一定的存储方式存储在计算机的存储器中,并在这些数据上定义一个运算的集合,就称为一个数据结构。

数据结构研究的内容如下。

- (1) 数据的逻辑结构:按照某种逻辑关系将数据组织好,即逻辑结构。
- (2) 数据的存储结构:将数据及数据之间的关系存储到存储区域中,即存储结构。
- (3) 数据的运算:在这些数据上定义一个基本运算的集合。

数据结构研究的三个方面的内容之间既有区别又有着密切的联系:数据的逻辑结构是从具体问题抽象出来的数学模型;数据的存储结构是逻辑结构到存储区域的映射;数据的运算是定义在数据的逻辑结构上的,而实现在存储结构上。

当用数据结构解决实际问题时,首先要根据具体问题的特点,归纳出数据的逻辑结构,并根据数据处理的需要,定义一个基本运算的集合。若要用计算机完成这些运算,必须将数据及数据之间的关系存放到计算机的存储器中,既实现数据的存储结构,然后才能编写程序实现定义的运算。有了逻辑结构及定义在逻辑结构上的运算,就解决了要“做什么?”的问题;而完成了存储结构及实现在存储结构上的运算,才真正解决了用计算机该“怎么做?”的问题。“做什么?”和“怎么做?”是计算机解决实际问题实现方案中的两个重要层次。

下面就从数据结构研究的三方面内容来给出常用的术语和概念。

1.2.1 数据的逻辑结构

数据的逻辑结构(Logical Structure):是数据元素之间的逻辑关系。它是根据实际问题本身所含数据之间的内在联系而抽象出来的数学模型,与计算机无关,所以被称为数据的逻辑结构。

由于数据的逻辑结构是数学模型,所以可以借助数学方法来表示,具体可以用离散数学中关系代数的二元组表示。

$\text{Data_Structure}=(D,S)$

$D=\{d_1,d_2,\dots,d_n\}$, $d_i(1\leq i\leq n)$ 代表数据元素。

$S=\{r_1,r_2,\dots,r_m\}$, $r_j(1\leq j\leq m)$ 代表数据元素之间的关系。

其中: D 是数据元素的有限集, S 是 D 上关系的有限集。若 S 为空集,则表示 D 中数据元素之间不存在关系,或者说是一种离散的关系,就像数学中集合里的元素一样。无关系的数据不作为本课程讨论的内容。当研究某一种数据结构时,通常取 S 中的一个关系 r_j 来进行讨论, r_j 可以表示为数据元素的序偶 $\langle d_i,d_j \rangle$ 的集合。如果集合中有序偶 $\langle d_i,d_j \rangle$,表示数据元素 d_i 和 d_j 之间有 r_j 这种关系。关于如何用二元组表示数据的逻辑结构,请看下面的例子。

例 1.1 某校围棋社团学生信息可以组织成表 1-1,其中每个学生为一个数据元素,共 8 个数据元素,每个数据元素包含 5 个数据项。

表 1-1 某校围棋社团学生信息表

学号	姓名	性别	出生日期	职务
201001001	黄家正	男	1992-08-05	团长
201002012	赵 芳	女	1993-08-15	组长
201003006	王 明	女	1993-04-01	组长
201008014	王 红	女	1992-06-28	组员
201102028	张小才	男	1994-03-17	组员
201103008	马立伟	男	1993-10-12	组员
201106007	孙 刚	男	1992-07-05	组员
201108019	刘 永	男	1992-12-09	组员

当组织好上面的信息表后,要处理的数据就具有了数据的逻辑结构,可以用二元组来表示这个逻辑结构。若用 d_1 表示表中第 1 个学生, d_2 表示第 2 个学生, $\dots$, d_8 表示第 8 个学生($n=8$)。 r_1 表示 8 名学生按学号从小到大顺序排列的关系,则表 1-1 的数学模型用二元组表示为:

$$L=(D,S),r_1\in S$$

$$D=\{d_1,d_2,d_3,d_4,d_5,d_6,d_7,d_8\}$$

$$r_1=\{\langle d_1,d_2 \rangle,\langle d_2,d_3 \rangle,\langle d_3,d_4 \rangle,\langle d_4,d_5 \rangle,\langle d_5,d_6 \rangle,\langle d_6,d_7 \rangle,\langle d_7,d_8 \rangle\}$$

此二元组表示的是数据结构中研究的最简单的逻辑结构,由于元素之间存在线性关系,通常称为线性表。

数据的逻辑结构还可以用更直观的图形方式来表示。若用小圆圈代表数据元素,小圆圈间的带箭头连线表示数据元素间的关系,则表 1-1 的逻辑结构可以表示为如图 1-1 所示。

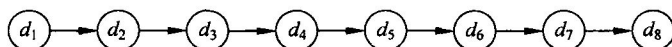


图 1-1 线性表的图示

针对用二元组表示的逻辑结构,有如下的常用术语。

(1) 前趋结点、后继结点、相邻结点

设 $\text{Data_Structure} = (D, S)$ 是一个逻辑结构, $d_i, d_j \in D, r_j \in S$, 若 $d_j < d_i, d_j > r_j$, 则称 d_i 是 d_j 相对 r_j 的前趋结点, d_j 是 d_i 相对 r_j 的后继结点, d_i 和 d_j 称为相邻结点。

(2) 开始结点、终端结点、内部结点

设 $\text{Data_Structure} = (D, S)$ 是一个逻辑结构, $d_i \in D, r_j \in S$, 若 d_i 相对 r_j 无前趋结点, 则称 d_i 是相对 r_j 的开始结点; 若 d_i 相对 r_j 无后继结点, 则称 d_i 是相对 r_j 的终端结点; 若 d_i 相对 r_j 既有前趋结点又有后继结点, 则称 d_i 是相对 r_j 的内部结点。

由于数据的逻辑结构是数据结构研究的三个方面内容的前提部分, 也就是说, 没有数据的逻辑结构就不可能有数据的存储结构, 也不可能有数据的运算。因此在不加严格区分的情况下, 常常将数据的逻辑结构简称为数据结构。数据结构通常分为两大类: 线性结构和非线性结构。

(1) 线性结构: 经典的线性结构是线性表。

线性结构的逻辑特征是: 有且仅有一个开始结点和一个终端结点, 其余的内部结点都有且仅有一个前趋结点和一个后继结点, 也就是说结构中的数据元素间存在着一对一的相互关系。比如例 1.1 中给出的线性表就具有线性结构的逻辑特征, 其中 d_1 是开始结点, d_n 是终端结点, 其余的都是内部结点。本书的第 2 章和第 3 章介绍的都是线性结构。

(2) 非线性结构: 经典的非线性结构有树形结构和图形结构。

树形结构的逻辑特征是: 有且仅有一个开始结点, 可有若干个终端结点, 其余的内部结点都有且仅有一个前趋结点, 可以有若干个后继结点, 也就是说结构中的数据元素间存在着一对多的层次关系。本书的第 5 章将介绍树形结构。

图形结构的逻辑特征是: 可有若干个开始结点和终端结点, 其余的内部结点可以有若干个前趋结点和若干个后继结点, 也就是说结构中的数据元素间存在着多对多的网状关系。本书的第 6 章将介绍图形结构。

下面的两个例子给出了树形结构和图形结构的二元组表示和图示。

例 1.2 在表 1-1 中, 学生之间还存在着领导与被领导的关系, 其中 d_1 为团长, 直接领导 d_2 和 d_3 , 而 d_2 和 d_3 是组长, d_2 直接领导 d_4, d_5 , d_3 直接领导 d_6, d_7, d_8 , 假设表示这种逻辑结构的关系为 r_2 , 则 r_2 可以定义为学生之间的领导与被领导关系, 其逻辑结构可用二元组表示。

$$T = (D, S), r_2 \in S$$

$$D = \{d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8\}$$

$$r_2 = \{ \langle d_1, d_2 \rangle, \langle d_1, d_3 \rangle, \langle d_2, d_4 \rangle, \langle d_2, d_5 \rangle, \\ \langle d_3, d_6 \rangle, \langle d_3, d_7 \rangle, \langle d_3, d_8 \rangle \}$$

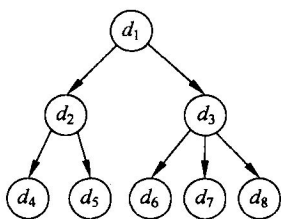


图 1-2 树形结构的图示

对应的逻辑结构图形表示如图 1-2 所示。可以看出, 此数据结构为树形结构, 与自然界的树相比, 它是一棵倒立的树。其中 d_1 是开始结点, 没有前趋; d_4, d_5, d_6, d_7, d_8 是终端结点, 没有后继; 其余的内部结点都只有一个前趋但有若干个后继。数据元素间是一对多的关系, 符合树形结构的逻辑特征。

例 1.3 在表 1-1 中, 学生之间还有好友关系, 假设将 r_3 定义为学生之间的好友关系, 该数据结构的逻辑结构可用二元组

表示。

$$G = (D, S), r_3 \in S$$

$$D = \{d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8\}$$

$$r_3 = \{ \langle d_1, d_2 \rangle, \langle d_2, d_1 \rangle, \langle d_1, d_3 \rangle, \langle d_3, d_1 \rangle, \langle d_1, d_5 \rangle, \langle d_5, d_1 \rangle, \\ \langle d_2, d_4 \rangle, \langle d_4, d_2 \rangle, \langle d_3, d_5 \rangle, \langle d_5, d_3 \rangle, \langle d_4, d_5 \rangle, \langle d_5, d_4 \rangle, \\ \langle d_4, d_6 \rangle, \langle d_6, d_4 \rangle, \langle d_6, d_7 \rangle, \langle d_7, d_6 \rangle \}$$

对应的逻辑结构图示如图 1-3 所示,其中 d_8 无好友,在图形中为孤立顶点。图形中每对顶点之间的两条相反的有向边,表示这两个顶点学生是一对好朋友。从图 1-3 可看出,数据元素间存在着多对多的关系,也就是说一个数据元素可以有任意多个前趋和任意多个后继,符合图形结构的逻辑特征。

从以上三个例子可以看出,虽然各例中数据元素是相同的,但是数据元素间的关系不同,所以它们是不同的数据结构。线性表、树和图是数据结构中重点研究的三种经典结构。

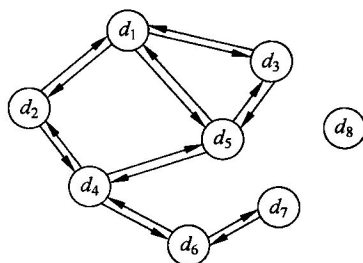


图 1-3 图形结构的图示

1.2.2 数据的存储结构

数据的存储结构(Storage Structure):是指数据的逻辑结构到计算机存储区域的映射。对于数据的逻辑结构 $\text{Data_Structure} = (D, S)$,在映射中,一方面要将数据集 D 中的数据元素存放到存储区域中,另一方面还要体现关系集 S ,常见的体现关系 S 的方式有显示和隐含两种。

计算机存储器是以字节(B)为单位进行编址的,地址是从零开始的、连续的正整数。每个字节包含 8 个二进制位,有唯一的整数地址标识,是存取操作的基本单位。用户使用存储器时通常是以存储单元为单位,每个存储单元由若干个连续的字节构成,一个单元可以存储一个数据元素,单元的大小取决于数据元素的类型。每个存储单元都有唯一的地址标识(即若干个连续字节的首地址),若干地址连续的存储单元称为一个存储区域,也可以说存储区域是存储单元的线性序列。用户可以通过每个存储单元的地址实现对存储单元中数据元素的访问。数据中的各个数据元素可以存储在连续的存储单元中,也可以存储在不连续的存储单元中。

完成数据的逻辑结构到存储区域的映射可以有很多的方法,常用的数据存储的方法有如下 4 种。

1. 顺序存储

顺序存储方法的基本思想是:把逻辑上相邻的数据元素存储在物理位置上相邻的存储单元里。数据元素间的逻辑关系由存储单元的邻接关系来体现,也就是说逻辑关系上相邻物理位置上也相邻,数据元素的逻辑次序与物理次序一致。这是一种隐含体现关系的存储方法,关系隐含在存储位置上。每个后继结点都存储在当前结点所在单元的下一个单元中。由此可以看出数据元素在存储区域中是连续存放的,这种存储方式称为顺序存储结构(Sequential Storage Structure),通常用计算机高级语言中的数组类型来描述。这种方法主要应用于线性结构,但非线性的数据结构也可以通过某种线性化方法实现顺序存储,如后面

将介绍的多维数组、二叉树等结构的顺序存储。

2. 链接存储

链接存储方法的基本思想是：通过附加指针域来表示数据元素之间的关系。这种存储方法不要求逻辑上相邻的数据元素存储位置上也相邻，数据元素间的逻辑关系是通过附加的指针(地址)来表示的，这是一种显示体现关系的存储方法。数据元素在存储区域中可以是连续的，也可以是不连续的，通常用计算机高级语言中的指针类型来描述，称为链接存储结构(Linked Storage Structure)。这种存储方法具有很大的灵活性，适用于大多数数据结构。但是与顺序存储方法相比，由于要附加指针域，所以它的空间开销要大一些。但由于不要求存储空间的连续性，很适合动态存储管理。

顺序和链接是两种最常用的存储方法，本书中从第2~7章研究各种经典结构时采用的都是这两种存储结构。

3. 索引存储

索引存储方法的基本思想是：通过附加索引表来表示数据元素之间的关系。索引表中的每一项称为索引项，用来标识一个或一组数据元素的存储位置。索引项一般形式为(关键字，地址)，其中的关键字是能唯一标识一个数据元素的数据项。若每个数据元素对应一个索引项，则该索引表为稠密索引(Dense Index)。若一组数据元素对应一个索引项，则该索引表称为稀疏索引(Sparse Index)。索引项中的地址指出数据元素的存储位置。索引存储方法主要是用于实现快速查找而设计的一种存储方式。

4. 散列存储

散列存储方法的基本思想是：根据数据元素的关键字直接计算出该数据元素的存储地址，通常称为关键字-地址转换法。在此方法中需要设计一个散列函数，以关键字为自变量，散列函数值即为地址。用这种存储方法设计的存储结构最适合按关键字进行查找，但数据元素之间的关系已经无法在存储结构上体现。

上述4种存储方法可以单独使用，也可以组合起来对数据结构进行存储。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。针对具体的应用，某种数据结构选择何种存储结构主要考虑运算的方便及效率。

如何描述存储结构呢？数据的存储结构是数据的逻辑结构用计算机语言的实现，它是依赖于计算机语言的，因此可以借用高级语言中提供的数据类型来描述它。本书采用C语言的数据类型来描述数据的存储结构，主要使用结构体、数组和指针等。

1.2.3 数据的运算

数据的运算(也称操作)是指对数据元素进行加工和处理。运算的种类很多，具体视应用的要求而设置。但对每种数据结构设置一些基本运算，使得其他运算都能通过调用这些基本操作实现对数据的各种访问，是数据结构中研究的一个重要内容。

数据的基本运算一般包括查找、插入、删除、排序等。这些基本运算的定义基于数据的逻辑结构，每种经典的逻辑结构都有一个运算的集合。至于基本运算的实现，只有确定了存储结构后才能完成，由此可以看出，数据的运算是定义在数据的逻辑结构上而实现在数据的存储结构上的。针对具体问题，首先要将待处理的数据组织好，就具有了数据的逻辑结构，然后在数据的逻辑结构上定义一个运算的集合，这时就解决了对具体的问题要处理些什么，