

21世纪高等学校计算机专业
核心课程规划教材

数据结构（C语言描述）

（第2版）

◎ 徐孝凯 编著



清华大学出版社

21世纪高等学校计算机专业
核心课程规划教材

数据结构 (C语言描述)

(第2版)

◎ 徐孝凯 编著

清华大学出版社
北京

内 容 简 介

本书是利用 C 语言编写的一本数据结构教材,适合在学习 C 语言之后使用。书中介绍了各种常用的数据结构、对应的存储结构,以及对其进行各种典型运算的方法和算法;并给出了丰富而实用的算法实例,这些实例都具有很好的可读性、结构化、模块化和时空有效性。通过深入学习和领会,能够大大提高读者的软件开发和设计能力。本书适合作为各级各类学校数据结构课程的教材或教学参考书,也适合软件开发人员参考。

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: C 语言描述/徐孝凯编著. —2 版. —北京:清华大学出版社,2018

(21 世纪高等学校计算机专业核心课程规划教材)

ISBN 978-7-302-49951-0

I. ①数… II. ①徐… 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 ②TP312.8

中国版本图书馆 CIP 数据核字(2018)第 066121 号

责任编辑:郑寅堃 薛 阳

封面设计:刘 键

责任校对:梁 毅

责任印制:丛怀宇

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:北京密云胶印厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:19.5

字 数:474 千字

版 次:2018 年 10 月第 2 版

印 次:2018 年 10 月第 1 次印刷

印 数:1~1500

定 价:59.00 元

产品编号:077223-01



第2版前言

数据结构是普通高等院校计算机及相关专业开设的一门主干基础课程,其教学内容主要包括数据的逻辑结构、存储(物理)结构及其相关运算。学习这门课程要解决的问题是:如何把现实世界中存在的大量数据,在逻辑和物理上有效地组织起来,形成一个数据系统,进而实施有关运算,从中得到人们所需要的有用信息。

数据结构课程所涉及的内容非常广泛而精深,各种数据组织和运算方法层出不穷,但如何能够让学生在有限的时间内学习到这门课程更多更好的知识体系,发挥出画龙点睛、举一反三的作用,这是每位编写此类教材的作者必须要面对和深思的问题。

本人在编写这本教材的第2版时,也进行了认真的思考,在众多数据结构教材中找准自己教材的定位,突出自己教材的特色。本教材具有以下鲜明特色。

(1) 本书内容具有很强的连贯性和系统性。

(2) 第1章是全书的起步和总览,简明扼要,开门见山。本书第1章通过具体数据处理示例,自然引出各种典型的数据结构,让读者对看似高深的概念一目了然,为深入学习后面各章做好准备。

(3) 本书对数据的各种逻辑结构和存储结构的讨论,采用统一的、规范化的方法,便于读者理解和掌握,并且还能举一反三、触类旁通。

(4) 本书对每种数据结构的运算,都采用抽象数据类型和抽象存储类型,定义出统一接口(函数原型声明),具体在对运算的算法描述(函数定义)中采用什么具体的存储类型,将与调用此函数的接口完全无关,这样就很好地体现了当今软件开发中所必须具有的多态性、封装性、隐藏性等特点,有利于读者养成良好的软件开发习惯和提高软件开发水平。

(5) 本书对每一种运算的方法分析和C语言算法描述都尽量做到条理清晰、循序渐进、深入浅出,并在算法内部的各语句中,都给出了简要注释和提示,力求使读者更容易理解和掌握。

(6) 每章后面都配有丰富的、各类题型的习题,给学习者充足的练习天地,以便巩固和加深理解所学知识。

本书若有幸被选作教材或教学参考书,建议读者:

(1) 要从头到尾、按照章节内容的先后次序、一步一个脚印地阅读,并在阅读过程中富于联想和思考,这样才能收到事半功倍的学习效果。

(2) 对于自己不明白或有疑问的算法,除了认真分析外,还要上机调试和运行,从实际上机操作中得到真实体验和收获。

(3) 在做每章后面的习题时,要尽量独立思考和完成,必要时进行同学之间的交流和讨论,或者向老师寻求帮助。无论做得对与错,都是对自己学习效果 and 能力的检验和提高,都是大有裨益的。千万不要急着寻找现成的答案。不经过自己反复思考而得到的结果是不巩固的。

为了给任课教师和自学读者提供方便,本人把教材中所有习题的参考解答交给编辑,从而放到清华大学出版社读者网站上供下载。另外,若愿意与本人交流,可发电子邮件联系, xuxk@crtvu.edu.cn, 还可通过编辑与笔者进行电话联系。

在本书构思和编写过程中,得到了中国石油大学陈明教授、首都师范大学王锁柱教授、中央民族大学曹永存教授、北京石油化工学院张晓明教授的热情关心、指导和帮助,本书的全部书稿得到了王锁柱教授不辞辛苦的审阅和修改,在此对各位专家表示真诚感谢!

由于本人学识水平有限,尽管做出了最大努力,但书中疏漏和不足之处在所难免,敬请专家和广大读者批评指正,本人不胜感激!

编者

2017年10月



第 1 版前言

数据结构主要研究数据的逻辑结构、在计算机中的存储结构,以及对数据进行各种非数值运算的方法和算法。数据的逻辑结构分为集合、线性、树(层次)、图(网状)4种基本结构,由它们可以构成任何所需要的较复杂的逻辑结构。数据的存储结构分为顺序、链接、索引、散列4种基本结构,同样由它们能够构成各种较复杂的存储结构。对数据进行的非数值运算主要包括查找、排序、插入、删除、修改、遍历等。对于同样的数据,若采用的逻辑结构和存储结构不同,对某一运算所采用的方法不同,则将得到不同的算法,进而在计算机上得到不同的运行时间和存储空间效率。通过该课程的学习,读者能够根据实际应用中数据处理的要求,为数据选择和建立合适的逻辑结构和存储结构,接着选择和使用较好的数据处理方法,以及利用一种程序设计语言编写出相应的算法,最后在计算机系统上调试、运行和实现算法。

本书是根据一般计算机及相关专业对开设数据结构课程的知识结构要求编写的,它介绍的是数据结构学科成熟而实用的知识,摒弃那些深奥难懂而又过时不用的内容;在写法上力求条理清楚、层次分明、内容连贯、循序渐进,便于阅读和自学;在各种运算方法和算法的分析上,力求细致、生动、深入、透彻,便于理解。

本书共包括9章。第1章概要介绍数据的逻辑结构和存储结构,以及算法的描述和评价。第2~4章主要介绍线性表、栈、队列、稀疏矩阵、广义表等线性数据结构的定义、存储结构,以及相应运算的方法和算法。第5、6章主要介绍树、二叉树、二叉搜索树、堆、哈夫曼树等树结构的定义和存储结构,以及建立、遍历、查找等运算的方法和算法。第7章主要介绍图的定义和存储结构,以及图的建立、遍历、求最小生成树、拓扑排序等运算的方法和算法。第8章主要介绍顺序、二分、索引、散列等查找的方法和算法。第9章主要介绍插入、选择、堆、快速、二路归并、外存文件等排序的方法和算法。

本书给出的所有算法和程序都在C语言运行环境下调试通过,同时,由于C++兼容C语言,本人也在VC++6.0环境下调试通过所有算法和程序,因此能够保证算法是正确和有效的。

本书每章后面都配有丰富的习题,同学们通过做练习和上机调试程序,能够巩固和提高所学知识,能够提高分析问题和解决问题的能力。由于数据结构习题一般较难,有时不太容易做出,为了使同学们得到帮助和受到启发,专门为本书配有习题参考解答一书,并同本书一同出版发行。当然,同学们要尽量独立思考完成作业,绝不能依赖现成的答案。

使用本书推荐讲授学时为64,上机实验学时为32,实验题目可采用教材中所给的调试



程序,也可由实验指导教师指定。数据结构课程所涉及的内容较多,而教学时数有限,所以各任课教师可根据实际情况对本书中的内容进行取舍。另外,在每章习题中,加星号的习题较难,可以不做要求。

本书由徐孝凯教授和贺桂英副教授共同编写,在教学大纲构思和教材内容编写的过程中得到了巫家敏教授的真诚指导,得到了徐培忠和郑寅堃的热情帮助,在此谨向他们表示衷心感谢!

由于作者水平有限,不足之处在所难免,敬请同行专家和广大读者指正。电子邮箱地址为 xuxk@crtvu.edu.cn。

徐孝凯

2004 年 2 月

目 录

第 1 章 绪论	1
1.1 基本概念	1
1.2 算法描述	10
1.3 算法评价	11
习题	18
第 2 章 集合	22
2.1 集合的定义和运算	22
2.2 集合的顺序存储结构和操作实现	23
2.3 集合的链接存储结构和操作实现	29
2.3.1 链接存储集合的概念	29
2.3.2 集合运算在链接存储结构下的操作实现	30
习题	35
第 3 章 线性表	40
3.1 线性表的定义和运算	40
3.2 线性表的顺序存储结构和操作实现	42
3.3 链接存储的一般概念和方法	49
3.4 线性表的链接存储结构和操作实现	53
习题	58
第 4 章 栈和队列	63
4.1 栈的定义和运算	63
4.2 栈的顺序存储结构和操作实现	64
4.3 栈的链接存储结构和操作实现	67
4.4 栈的简单应用举例	70
4.5 算术表达式的计算	74
4.5.1 算术表达式的两种表示	74

4.5.2	后缀表达式求值的算法	75
4.5.3	把中缀表达式转换为后缀表达式的算法	78
4.6	栈与递归	81
4.7	队列	89
4.7.1	队列的定义和运算	89
4.7.2	队列的顺序存储结构和操作实现	90
4.7.3	队列的链接存储结构和操作实现	94
4.7.4	队列的应用简介	96
习题	97
第 5 章	树和二叉树	102
5.1	树的概念	102
5.1.1	树的定义	102
5.1.2	树的表示	104
5.1.3	树的基本术语	104
5.1.4	树的性质	105
5.2	二叉树	106
5.2.1	二叉树的定义	106
5.2.2	二叉树的性质	107
5.2.3	二叉树的抽象数据类型	108
5.2.4	二叉树的存储结构	109
5.3	二叉树遍历	112
5.4	二叉树其他运算	114
5.5	树的存储结构和运算	124
5.5.1	树的抽象数据类型	124
5.5.2	树的存储结构	124
5.5.3	树的运算	125
习题	132
第 6 章	二叉树应用	137
6.1	二叉搜索树	137
6.1.1	二叉搜索树的定义	137
6.1.2	二叉搜索树的抽象数据类型	138
6.1.3	二叉搜索树的运算	138
6.2	堆	145
6.2.1	堆的定义	145
6.2.2	堆的抽象数据类型	146
6.2.3	堆的存储结构	147
6.2.4	堆的运算	148

6.3 哈夫曼树	152
6.3.1 基本术语	152
6.3.2 构造哈夫曼树	153
6.3.3 哈夫曼编码	155
习题	158
第7章 图	162
7.1 图的概念	162
7.1.1 图的定义	162
7.1.2 图的基本术语	163
7.1.3 图的抽象数据类型	165
7.2 图的存储结构	166
7.2.1 邻接矩阵	166
7.2.2 邻接表	169
7.2.3 边集数组	172
7.3 图的遍历	173
7.3.1 深度优先搜索遍历	174
7.3.2 广度优先搜索遍历	177
7.3.3 非连通图的遍历	180
7.3.4 图的遍历算法的上机调试	180
7.4 图的其他运算	183
习题	194
第8章 图的应用	198
8.1 图的生成树和最小生成树	198
8.1.1 生成树和最小生成树的概念	198
8.1.2 普里姆算法	200
8.1.3 克鲁斯卡尔算法	204
8.2 最短路径	207
8.2.1 最短路径的概念	207
8.2.2 从图中一顶点到其余各顶点的最短路径	208
8.2.3 图中每对顶点之间的最短路径	214
8.3 拓扑排序	217
8.3.1 拓扑排序的概念	217
8.3.2 拓扑排序算法	219
8.4 关键路径	223
8.4.1 顶点事件的发生时间	223
8.4.2 计算关键路径的方法和算法	226
习题	230



第 9 章 查找	234
9.1 查找的概念	234
9.2 顺序表查找	235
9.2.1 顺序查找	235
9.2.2 二分查找	237
9.3 索引查找	241
9.3.1 索引的概念	241
9.3.2 索引查找算法	244
9.3.3 分块查找	246
9.4 散列查找	248
9.4.1 散列的概念	248
9.4.2 散列函数	249
9.4.3 处理冲突的方法	251
9.4.4 散列表的运算	254
9.5 B 树查找	261
9.5.1 B 树定义	261
9.5.2 B 树查找	263
9.5.3 B 树插入	264
9.5.4 B 树删除	265
习题	268
第 10 章 排序	272
10.1 排序的基本概念	272
10.2 插入排序	274
10.2.1 直接插入排序	274
10.2.2 希尔排序	274
10.3 选择排序	276
10.3.1 直接选择排序	276
10.3.2 堆排序	277
10.4 交换排序	281
10.4.1 气泡排序	281
10.4.2 快速排序	282
10.5 归并排序	285
10.6 各种内排序方法的比较	289
10.7 外排序	290
10.7.1 外排序概念	290
10.7.2 外排序算法	292
习题	297

1.1 基本概念

数据结构是计算机、信息管理等相关专业的一门专业基础课程,它专门讨论由人们根据解决问题的需要,把从现实世界中抽象出来的数据,如何在计算机中进行有效地表示、存取和处理。这里所说的数据是广义的,它不仅包括数值数据、字符数据、日期数据等简单数据,而且包括带有一定结构的各种复杂数据,如记录数据、向量数据、矩阵数据、表格数据、图形数据、图像数据、音频数据、视频数据等。

用计算机存储数据不仅要存储数据本身,而且要存储数据之间的联系(即数据结构),使得解决一个问题的所有数据集变为一个有机的数据整体,这样就能够根据一定的路径和搜索方法快速地访问到所需要的任何数据元素,并有效地利用。在计算机中存储数据有许多具体的方法,每种方法都称为数据的一种存储结构,通常可分类为顺序存储结构、链接存储结构、散列存储结构、索引存储结构等,当然这些都是存储数据的最基本结构,由它们组合可以形成较为复杂的各种存储结构。

对数据进行处理的方法称为算法,它是根据人们解决实际问题的需要而逐渐产生、发展和丰富起来的。到目前为止,人们已经总结出许多常用和有效的处理数据的算法,了解、掌握和使用这些算法是数据结构课程的主要内容。这将为学习后续专业课程,以及进行软件开发和设计奠定坚实基础。进行数据处理的算法就是为解决问题而设计的具体方法和步骤,若要让一个算法在计算机上真正实现(执行),从而得到预期的结果,就必须事先学习和掌握好一种计算机语言,如 C、C++、Java 语言等,利用计算机语言作为工具,编写出描述算法的相应代码(程序),然后再通过上机调试和运行,就可以实现预期的结果。本书拟采用的计算机算法描述语言是高校中最常用的 C 语言,每个算法描述的过程(函数、模块)都可以在 C 语言编译环境下调试和运行,从而能够检查和验证算法的正确性和有效性。

下面就本教材中经常使用到的一些术语进行解释,以便后续使用。

1. 数据

数据(data)是人们利用文字符号、数字符号以及其他规定的符号对现实世界的事物及其活动所做的抽象描述。例如,一个人的名字可以用一个字符串来描述,一条曲线可以用一个数组来描述,依次取出曲线中的不同坐标点,数组中的每一元素用来对应存储每个点的坐标值和颜色编号。因此,一个文档、记录、数组、句子、单词、算式、数值、字符等都统称为数据。在计算机领域,人们把能够被计算机处理的对象,或者说能够被计算机输入、存储、计算、加工、输出的一切信息都叫做数据。

2. 数据元素

数据元素(data element)简称**元素**,它是一个数据整体中相对独立的单位。如对于一个

数据文件来说,所存储的每个记录就是它的数据元素;对于一个字符串来说,其中的每个字符就是它的数据元素;对于一个数组来说,每个下标位置上的表示值就是它的数据元素。数据和数据元素是相对而言的,是整体和个体的关系。如对于一个记录数据来说,它是所属文件这个整体数据中的一个数据元素,而它相对于所含的数据项而言又是一个整体数据,数据项是记录中的个体,是记录中的一个成分(元素)。因此,在叙述中很难对数据和数据元素进行严格区别,读者应根据上下文理解其含义。

3. 数据记录

数据记录(data record)简称**记录**,它是数据处理领域组织数据的基本单位,数据中的每个数据元素在许多应用场合都被组织成记录结构。一个数据记录由一个或多个**数据项**(item)所组成,每个数据项可以是简单数据项(即不可再分,如一个数值、一个字符等),也可以是组合数据项(即数组或记录等)。就拿对图书目录管理来说,每个记录表示一本图书的有关信息,如表 1-1 所示。

表 1-1 图书目录表

登录号	书号	书名	作者	出版社	定价
00001	978-7-302-15761-8	C++ 语言基础教程	徐孝凯	清华大学	35.00
00002	978-7-302-18160-6	计算机导论	陈明	清华大学	28.00
00003	978-7-111-25859-9	数据库原理及应用	吴靖	机械工业	29.00
00004	978-7-304-04066-6	C 语言程序设计	任爱华	中央电大	21.00
00005	978-7-5618-2646-1	动态网页制作实例教程	丁桂芝	天津大学	41.00
00006	978-75640-7819-5	网络影视课件学教程	李小平	北京理工大学	39.00
00007	978-7-5005-8441-4	网络数据库技术	李红	中国财政经济	24.00
⋮					

在表 1-1 中,第一行为表目行或目录行,它给出了书目表中每条记录的组成(结构)。从表目行向下的每一行为一条记录,它给出了一册图书的有关登记信息;每一列为一个数据项,它描述了图书目录中的一种属性。每条记录由 6 个数据项组成,其名称分别为登录号、书号、书名、作者、出版社和定价,前 5 个数据项均为字符串类型,后一个数据项为实数类型。

在一个表中,若所有记录在某个数据项中的值均不同,也就是说,每个值能够唯一地标识一个记录时,则可把这个数据项定义为记录的**关键数据项**,简称**关键项**(key item),关键项中的每一个值称为所在记录的**关键字**(key word 或 key)。如在表 1-1 中,登录号数据项中的所有值均不同,所以可把登录号作为记录的关键项,其中的每一个值就是所在记录的关键字。如 00002 为第 2 条记录的关键字,00005 为第 5 条记录的关键字。

在一个表中,能作为关键项的数据项可能没有,可能只有一个,也可能多于一个。当没有时,可把多个有关的数据项联合起来,构成一个**组合关键项**,用组合关键项中的每一个组合值来唯一地标识一个记录,该组合值就是所在记录的关键字。

引入了记录的关键项和关键字后,为简便起见,在以后的讨论中,经常利用关键项来标识所有记录,利用关键字来标识所在的记录,而把记录中的其他项忽略掉。如表 1-1 可以简记为(00001,00002,00003,00004,00005,00006,00007,⋯),第 2 条记录可以简记为 00002。

4. 数据结构

数据结构(data structure)是指数据及其相互之间的联系。上面已经提到过,数据的描

述对象是现实世界的事物及其活动,而任何事物及其活动都不是孤立存在的,都是在一定意义上相互联系、相互影响的,所以数据之间必然存在联系,特别是具有相同性质的数据更是如此。数据之间的相互联系,被确切地称为数据的**逻辑结构**(logical structure)。在计算机中存储数据时,不仅要存储数据本身,而且要存储它们之间的联系(即逻辑结构)。数据在计算机存储器中的存储表示,被称为数据的**物理结构或存储结构**(storage structure),它是数据逻辑结构的存储映像。由于数据的存储结构具有顺序、链接、索引、散列等各种不同的方式,所以,一种数据结构(即逻辑结构)可以根据数据处理的需要选用任一种相应的存储结构被存储到计算机系统中。数据的逻辑结构是在现实世界的层面上反映出的数据结构,而数据的存储结构是在计算机世界的层面上反映出的数据结构,它们时常都被简称为数据结构,读者应根据上下文理解其具体含义。

为了更确切地描述数据的逻辑结构,通常采用二元组表示:

$$B = (K, R)$$

B 表示一种数据结构,它由数据元素的集合 K 和 K 上二元关系的集合 R 所组成。其中

$$K = \{k_i \mid 1 \leq i \leq n, n \geq 0\}$$

$$R = \{r_j \mid 1 \leq j \leq m, m \geq 0\}$$

其中, k_i 表示集合 K 中的第 i 个数据元素, n 为 K 中数据元素的个数,特别地,若 $n=0$, 则 K 是一个空集,没有任何元素,此时 B 也就无结构而言,有时也可以认为它具有任一结构; r_j 表示集合 R 中的第 j 个二元关系(以后简称为关系), m 为 R 中关系的个数,特别地,若 $m=0$, 则 R 是一个空集,表明不考虑集合 K 中元素之间是否存在任何关系,元素之间彼此是独立的,没有次序关系,就像数学中集合里的元素一样。在本书所讨论的数据结构中,一般只讨论 $m=1$ 的情况,即 R 中只包含一个关系($R=\{r_1\}$),并且直接把这个关系用 R 表示。若有些数据结构中包含多个关系,可分别对每一个关系进行讨论。

K 上的一个关系 R 是序偶的集合。对于 R 中的任一序偶 $\langle x, y \rangle (x, y \in K)$, 我们把 x 叫做序偶的第一元素,把 y 叫做序偶的第二元素,又称序偶的第一元素为第二元素的**直接前驱**(为叙述简便后面简称为前驱),称第二元素为第一元素的**直接后继**(简称后继)。如在 $\langle x, y \rangle$ 的序偶中, x 为 y 的前驱,确切地说为直接前驱,而 y 为 x 的后继,确切地说为直接后继。

一种数据结构还能够利用图形示意地表示出来,示意图中的每个结点(或叫顶点)对应着一个数据元素,两结点之间带箭头的连线(又称做有向边或弧)对应着关系中的一个序偶,其中,序偶的第一元素为有向边的起始结点,第二元素为有向边的终止结点,即箭头所指向的结点。

作为例子,根据表 1-2 构造出一些典型的数据结构。

表 1-2 教务处人事简表

职工号	姓名	性别	出生日期	职务	部门
01	江明华	男	1968.03.20	处长	
02	刘宁	男	1976.06.14	主任	办公室
03	张利	女	1972.12.07	科长	学籍科

续表

职工号	姓名	性别	出生日期	职务	部门
04	仝婷婷	女	1980.08.05	科长	考务科
05	刘永年	男	1967.08.15	科员	办公室
06	王明理	女	1983.04.01	科员	学籍科
07	鲍娟	女	1980.06.28	科员	学籍科
08	张才	男	1975.03.17	科员	考务科
09	陈书琴	女	1985.10.12	科员	考务科
10	徐宗英	男	1988.07.05	科员	考务科

表 1-2 中共有 10 条记录, 每条记录都由 6 个数据项所组成, 由于每条记录的职工号各不相同, 所以可把每条记录的职工号作为该记录的关键字, 并在下面的例子中, 我们将用记录的关键字来代表整个记录。

例 1-1 一种数据结构 $set=(K,R)$, 其中

$$K = \{01,02,03,04,05,06,07,08,09,10\}$$

$$R = \{\}$$

在这种数据结构 set 中, 只存在着元素的集合, 不存在有关系的集合, 表明我们只考虑表 1-2 中的每条记录, 并不考虑它们之间的任何关系。我们称具有此种特点的数据结构为**集合结构**(set structure)。对于集合结构, 元素之间按任何次序排列都是允许的, 如可以按关键字的升序排列, 也可以按关键字的降序排列, 等等, 这可根据处理问题的需要任意决定。

例 1-2 一种数据结构 $linear=(K,R)$, 其中

$$K = \{01,02,03,04,05,06,07,08,09,10\}$$

$$R = \{<05,01>, <01,03>, <03,08>, <08,02>, <02,07>, \\ <07,04>, <04,06>, <06,09>, <09,10>\}$$

对应的图形如图 1-1 所示。



图 1-1 数据的线性结构示意图

结合表 1-2, 细心的读者不难理解: R 是按职工年龄从大到小排列的线性关系。

在这种数据结构 $linear$ 中, 每个数据元素有且仅有一个直接前驱元素(除结构中第一个元素 05 外), 有且仅有一个直接后继元素(除结构中最后一个元素 10 外)。这种数据结构的特点是数据元素之间的 1 对 1(1:1)联系, 由此构成**线性关系**。我们把具有这种特点的数据结构叫做**线性结构**(linear structure)。

例 1-3 一种数据结构 $tree=(K,R)$, 其中

$$K = \{01,02,03,04,05,06,07,08,09,10\}$$

$$R = \{<01,02>, <01,03>, <01,04>, <02,05>, \\ <03,06>, <03,07>, <04,08>, <04,09>, <04,10>\}$$

对应的示意图如图 1-2 所示。

结合表 1-2, 细心的读者不难看出: R 是人员之间领导与被领导的关系。

图 1-2 像倒着画的一棵树, 在这棵树中, 最上面一层的一个没有前驱只有后继的结点叫

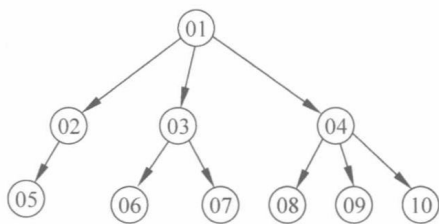


图 1-2 数据的树结构示意图

做树根结点,最下面一层的只有前驱没有后继的结点叫做树叶结点,除树根和树叶之外的结点叫做树枝结点。在一棵树中,每个结点有且只有一个前驱结点(除树根结点外),但可以有任意多个后继结点(树叶结点可看作具有 0 个后继结点)。这种数据结构的特点是数据元素之间的 1 对 $N(1:N)$ 联系($N \geq 0$),从而形成结点之间的层次关系。我们把具有这种特点的数据结构叫做树结构(tree structure),简称树。

例 1-4 一种数据结构 $\text{graph}=(K,R)$,其中

$$K = \{01,02,03,04,05,06,07,08,09,10\}$$

$$R = \{ \langle 01,02 \rangle, \langle 02,01 \rangle, \langle 01,03 \rangle, \langle 03,01 \rangle, \langle 01,06 \rangle, \langle 06,01 \rangle, \\ \langle 02,05 \rangle, \langle 05,02 \rangle, \langle 03,05 \rangle, \langle 05,03 \rangle, \langle 03,04 \rangle, \langle 04,03 \rangle, \\ \langle 03,08 \rangle, \langle 08,03 \rangle, \langle 06,08 \rangle, \langle 08,06 \rangle, \langle 05,07 \rangle, \langle 07,05 \rangle, \\ \langle 04,09 \rangle, \langle 09,04 \rangle, \langle 08,10 \rangle, \langle 10,08 \rangle \}$$

对应的图形如图 1-3 所示。

从图 1-3 可以看出, R 是 K 上的对称关系,为了简化起见,我们把 $\langle x,y \rangle$ 和 $\langle y,x \rangle$ 这两个对称序偶用一个无序对 (x,y) 或 (y,x) 来代替;在示意图中,我们把 x 结点和 y 结点之间两条相反的有向边用一条无向边来代替。这样 R 关系可改写为:

$$R = \{ (01,02), (01,03), (01,06), (02,05), (03,05), (03,04), \\ (03,08), (05,07), (04,09), (06,08), (08,10) \}$$

对应的图形如图 1-4 所示。

如果说 R 中每个序偶里的两个元素所代表的人员是好友,那么 R 关系就是人员之间的好友关系。

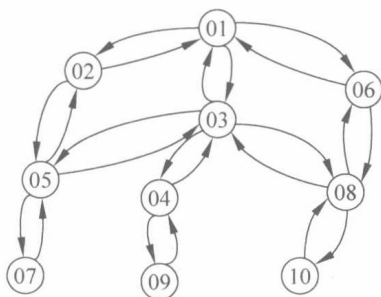


图 1-3 数据的图结构示意图

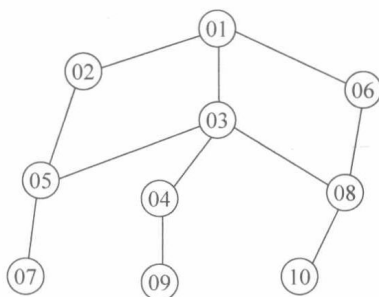


图 1-4 图 1-3 的等价表示

从图 1-3 或图 1-4 可以看出,结点之间的联系是 M 对 $N(M:N)$ 联系($M \geq 0, N \geq 0$),即网状关系,也就是说,每个结点可以有任意多个前驱结点和任意多个后继结点。我们把具有这种特点的数据结构叫做图结构或网结构(graph/chain structure),简称图或网。

由图结构、树结构和线性结构的定义可知,树结构是图结构的特殊情况,线性结构又是树结构的特殊情况,当然更是图结构的特殊情况。为了区别于线性结构,我们把树结构和图结构统称为**非线性结构**。线性结构和非线性结构相对于集合来说都是有序结构的,而集合是无序结构,即元素之间无次序。

例 1-5 一种数据结构 $B=(K,R)$, 其中

$$K = \{k_1, k_2, k_3, k_4, k_5, k_6\}$$

$$R = \{r_1, r_2\}$$

$$r_1 = \{\langle k_3, k_2 \rangle, \langle k_3, k_5 \rangle, \langle k_2, k_1 \rangle, \langle k_5, k_4 \rangle, \langle k_5, k_6 \rangle\}$$

$$r_2 = \{\langle k_1, k_2 \rangle, \langle k_2, k_3 \rangle, \langle k_3, k_4 \rangle, \langle k_4, k_5 \rangle, \langle k_5, k_6 \rangle\}$$

若用实线表示关系 r_1 , 虚线表示关系 r_2 , 则对应的图形如图 1-5 所示。

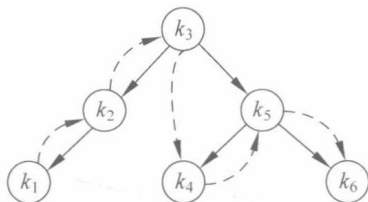


图 1-5 带有两个关系的一种数据结构示意图

从图 1-5 可以看出:此种数据结构 B 是一种非线性的图结构。但是,若只考虑关系 r_1 则为树结构,若只考虑关系 r_2 , 则为线性结构。

当一组数据中包含多个二元关系时,通常把它们看作多个数据结构来分别进行分析和处理。如对于上述如表 1-2 所示的教务处人事简表,根据处理不同问题的需要,就可能存在着按年龄排序建立的线性关系,按领导和被领导建立的层次关系,按好友联系建立的网状关系,等等。从而在同一个数据集合上能够分别构建成适应各种不同用途的数据结构。

5. 数据类型

数据类型(data type)是对数据的取值范围、数据结构以及允许施加运算的整体描述。在每一种计算机语言中都定义有自己的标准数据类型,通常都定义有整数、实数(浮点数)、枚举、字符、字符串、指针、数组、记录、文件等数据类型。如整数类型在计算机系统中通常用两个字节或四个字节表示,若采用两个字节,则整数表示范围在 $-2^{15} \sim 2^{15} - 1$, 即 $-32\,768 \sim 32\,767$; 若采用四个字节,则整数表示范围在 $-2^{31} \sim 2^{31} - 1$, 即 $-2\,147\,483\,648 \sim 2\,147\,483\,647$ 。整型数据之间的关系就是从小到大有序的线性关系,整型数据结构就是线性结构。对整型类型的数据允许施加的操作通常有:单目取正或取负运算,双目加、减、乘、除、取模等运算,双目等于、不等于、大于、大于等于、小于、小于等于等关系(比较)运算,以及赋值运算。字符类型在机器中通常用一个字节或两个字节表示,当使用一个字节时,无符号表示范围在 $0 \sim 255$ 之间,能够至多对 256 种字符进行识别、编码和存储,字符类型数据之间的关系是按 ASCII 码表中的字符编码有序,也是线性关系,字符数据结构也是线性结构,对字符类型的数据允许进行的操作主要为赋值和各种关系(比较)运算。字符串类型为字符的组合类型,其中的每个值都是一个字符的有限序列,字符之间是前后按位置有序的线性结构,对字符串的运算(操作)主要有求串长度、串拷贝、两串连接、两串比较等。

数据类型可分为简单类型和组合类型两种。简单类型中的每个数据即为简单数据,如