

数据结构 案例教程(C语言版)

主 编 徐翠霞
副主编 崔玲玲 邵回祖



北京大学出版社
PEKING UNIVERSITY PRESS

21 世纪全国高职高专计算机案例型规划教材

数据结构案例教程(C 语言版)

主 编	徐翠霞	
副主编	崔玲玲	邵回祖
参 编	周 彬	杨 平
	欧林梅	陈翠松



北京大学出版社
PEKING UNIVERSITY PRESS

内 容 简 介

本书系统地介绍了各种常用的数据结构,内容丰富,概念讲解清楚,叙述严谨流畅,逻辑性强。书中配备了大量的案例,每个案例都经过精心设计,既能帮助读者理解知识,又具有启发性。本书对给出的每一种算法,均先描述了它的基本思路和要点,使得算法清晰易读,便于学生理解和掌握。

本书共分为8章,内容包括线性表,栈和队列,串、数组和广义表,树和二叉树,图,查找,排序和综合实训。

本书可作为高职高专院校计算机类专业或信息类专业的教材,也可供从事计算机工程与应用的广大科技工作者参考。

图书在版编目(CIP)数据

数据结构案例教程(C语言版)/徐翠霞主编. —北京:北京大学出版社, 2009.3

(21世纪全国高职高专计算机案例型规划教材)

ISBN 978-7-301-14463-3

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

中国版本图书馆CIP数据核字(2009)第005907号

书 名: 数据结构案例教程(C语言版)

著作责任者: 徐翠霞 主编

策 划 编 辑: 李彦红 王显超

责 任 编 辑: 魏红梅

标 准 书 号: ISBN 978-7-301-14463-3/TP·0979

出 版 者: 北京大学出版社

地 址: 北京市海淀区成府路205号 100871

网 址: <http://www.pup.cn> <http://www.pup6.com>

电 话: 邮购部 62752015 发行部 62750672 编辑部 62750667 出版部 62754962

电 子 邮 箱: pup_6@163.com

印 刷 者: 北京山润国际印务有限公司

发 行 者: 北京大学出版社

经 销 者: 新华书店

787mm×1092mm 16开本 18.25印张 417千字

2009年3月第1版 2009年3月第1次印刷

定 价: 28.00元

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有 侵权必究

举报电话: 010-62752024

电子邮箱: fd@pup.pku.edu.cn

21 世纪全国高职高专计算机案例型规划教材 专家编写指导委员会

主 任 刘瑞挺 南开大学

副主任 安志远 北华航天工业学院

 丁桂芝 天津职业大学

委 员 (按拼音顺序排名)

陈 平 马鞍山师范高等专科学校

褚建立 邢台职业技术学院

付忠勇 北京政法职业技术学院

高爱国 淄博职业学院

黄金波 辽宁工程技术大学职业技术学院

李 纓 中华女子学院山东分院

李文华 湖北仙桃职业技术学院

李英兰 西北大学软件职业技术学院

田启明 温州职业技术学院

王成端 潍坊学院

王凤华 唐山工业职业技术学院

薛铁鹰 北京农业职业技术学院

张怀中 湖北职业技术学院

张秀玉 福建信息职业技术学院

赵俊生 甘肃省合作民族师范高等专科学校

周 奇 广东新安职业技术学院

顾 问 马 力 微软(中国)公司 Office 软件资深教师

 王立军 教育部教育管理信息中心

信息技术的案例型教材建设

(代丛书序)

刘瑞挺

北京大学出版社第六事业部在 2005 年组织编写了《21 世纪全国应用型本科计算机系列实用规划教材》，至今已出版了 50 多种。这些教材出版后，在全国高校引起热烈反响，可谓初战告捷。这使北京大学出版社的计算机教材市场规模迅速扩大，编辑队伍茁壮成长，经济效益明显增强，与各类高校师生的关系更加密切。

2008 年 1 月北京大学出版社第六事业部在北京召开了“21 世纪全国应用型本科计算机案例型教材建设和教学研讨会”。这次会议为编写案例型教材做了深入的探讨和具体的部署，制定了详细的编写目的、丛书特色、内容要求和风格规范。在内容上强调面向应用、能力驱动、精选案例、严把质量；在风格上力求文字精练、脉络清晰、图表明快、版式新颖。这次会议吹响了提高教材质量第二战役的进军号。

案例型教材真能提高教学的质量吗？

是的。著名法国哲学家、数学家勒内·笛卡儿(Rene Descartes, 1596—1650)说得好：“由一个例子的考察，我们可以抽出一条规律。(From the consideration of an example we can form a rule.)”事实上，他发明的直角坐标系，正是通过生活实例而得到的灵感。据说是在 1619 年夏天，笛卡儿因病住进医院。中午他躺在病床上，苦苦思索一个数学问题时，忽然看到天花板上有一只苍蝇飞来飞去。当时天花板是用木条做成正方形的格子。笛卡儿发现，要说出这只苍蝇在天花板上的位置，只需说出苍蝇在天花板上的第几行和第几列。当苍蝇落在第四行、第五列的那个正方形时，可以用(4, 5)来表示这个位置……由此他联想到可用类似的办法来描述一个点在平面上的位置。他高兴地跳下床，喊着“我找到了，找到了”，然而不小心把国际象棋撒了一地。当他的目光落到棋盘上时，又兴奋地一拍大腿：“对，对，就是这个图”。笛卡儿锲而不舍的毅力，苦思冥想的钻研，使他开创了解析几何的新纪元。千百年来，代数与几何，井水不犯河水。17 世纪后，数学突飞猛进的发展，在很大程度上归功于笛卡儿坐标系和解析几何学的创立。

这个故事，听起来与阿基米德在浴池洗澡而发现浮力原理，牛顿在苹果树下遇到苹果落到头上而发现万有引力定律，确有异曲同工之妙。这就证明，一个好的例子往往能激发灵感，由特殊到一般，联想出普遍的规律，即所谓的“一叶知秋”、“见微知著”的意思。

回顾计算机发明的历史，每一台机器、每一颗芯片、每一种操作系统、每一类编程语言、每一个算法、每一套软件、每一款外部设备，无不像闪光的珍珠串在一起。每个案例都闪烁着智慧的火花，是创新思想不竭的源泉。在计算机科学技术领域，这样的案例就像大海岸边的贝壳，俯拾皆是。

事实上，案例研究(Case Study)是现代科学广泛使用的一种方法。Case 包含的意义很广：包括 Example 例子，Instance 事例、示例，Actual State 实际状况，Circumstance 情况、事件、境遇，甚至 Project 项目、工程等。

我们知道在计算机的科学术语中，很多是直接来自日常生活的。例如 Computer 一词早在 1646 年就出现于古代英文字典中，但当时它的意义不是“计算机”而是“计算工人”，

即专门从事简单计算的工人。同理, Printer 当时也是“印刷工人”而不是“打印机”。正是由于这些“计算工人”和“印刷工人”常出现计算错误和印刷错误,才激发查尔斯·巴贝奇(Charles Babbage, 1791—1871)设计了差分机和分析机,这是最早的专用计算机和通用计算机。这位英国剑桥大学数学教授、机械设计专家、经济学家和哲学家是国际公认的“计算机之父”。

20 世纪 40 年代,人们还用 Calculator 表示计算机。到电子计算机出现后,才用 Computer 表示计算机。此外,硬件(Hardware)和软件(Software)来自销售人员。总线(Bus)就是公共汽车或大巴,故障和排除故障源自格瑞斯·霍普(Grace Hopper, 1906—1992)发现的“飞蛾子”(Bug)和“抓蛾子”或“抓虫子”(Debug)。其他如鼠标、菜单……不胜枚举。至于哲学家进餐问题,理发师睡觉问题更是操作系统文化中脍炙人口的经典。

以计算机为核心的信息技术,从一开始就与应用紧密结合。例如,ENIAC 用于弹道曲线的计算,ARPANET 用于资源共享以及核战争时的可靠通信。即使是非常抽象的图灵机模型,也受到二战时图灵博士破译纳粹密码工作的影响。

在信息技术中,既有许多成功的案例,也有不少失败的案例;既有先成功而后失败的案例,也有先失败而后成功的案例。好好研究它们的成功经验和失败教训,对于编写案例型教材有重要的意义。

我国正在实现中华民族的伟大复兴,教育是民族振兴的基石。改革开放 30 年来,我国高等教育在数量上、规模上已有相当的发展。当前的任务是提高培养人才的质量,必须从学科知识的灌输转变为素质与能力的培养。应当指出,大学课堂在高新技术的武装下,利用 PPT 进行的“高速灌输”、“翻页宣科”有愈演愈烈的趋势,我们不能容忍用“技术”绑架教学,而是让教学工作乘信息技术的东风自由地飞翔。

本系列教材的编写,以学生就业所需的专业知识和操作技能为着眼点,在适度的基础知识与理论体系覆盖下,突出应用型、技能型教学的实用性和可操作性,强化案例教学。本套教材将会有机融入大量最新的示例、实例以及操作性较强的案例,力求提高教材的趣味性和实用性,打破传统教材自身知识框架的封闭性,强化实际操作的训练,使本系列教材做到“教师易教,学生乐学,技能实用”。有了广阔的应用背景,再造计算机案例型教材就有了基础。

我相信北京大学出版社在全国各地高校教师的积极支持下,精心设计,严格把关,一定能够建设出一批符合计算机应用型人才培养模式的、以案例型为创新点和兴奋点的精品教材,并且通过一体化设计、实现多种媒体有机结合的立体化教材,为各门计算机课程配齐电子教案、学习指导、习题解答、课程设计等辅导资料。让我们用锲而不舍的毅力,勤奋好学的钻研,向着共同的目标努力吧!

刘瑞挺教授 本系列教材编写指导委员会主任、全国高等院校计算机基础教育研究会副会长、中国计算机学会普及工作委员会顾问、教育部考试中心全国计算机应用技术证书考试委员会副主任、全国计算机等级考试顾问。曾任教育部理科计算机科学教学指导委员会委员、中国计算机学会教育培训委员会副主任。PC Magazine《个人电脑》总编辑、CHIP《新电脑》总顾问、清华大学《计算机教育》总策划。

前 言

在计算机及其应用的各个领域中会用到各种各样的数据结构,学会分析研究计算机加工对象的特性,选择合适的数据结构和存储表示方法,以及编写相应的实现算法,是计算机工作者必不可少的技能。因此,数据结构是高等院校计算机专业教学中的一门重要专业基础课。在我国当前的计算机专业教学计划中,它是核心课程之一。

目前市面上的大多数数据结构教材都是传统编排模式:先是入门总体介绍,然后是基础知识,接着是算法体系等。这种方法以知识点为主线,过于拘泥于语法细节,而忽略了程序本身的重要性。

基于案例教学过程的实践和思考,更是为了培养读者的编程能力,编者提出了一种新的编写思路,打破传统教材的编写模式,先从案例入手,提出解决问题的方法和思路,分析问题需要的知识点,然后根据需要讲解知识点,再解决提出的问题,最后举一反三并以应用实例加以巩固和提升知识点,实现综合应用的目的。这样的教材或参考书目市面上很少,编者经过不断探讨并结合多年的案例教学经验,最终完成了本书的编写工作。

案例教学是计算机语言教学中最有效的方法之一,好的案例对学生理解知识、掌握如何应用知识都十分重要。本书以指导案例教学为目的,围绕教学内容组织案例,对学生的知识和能力训练具有很强的针对性,主要特色如下。

- (1) 以知识为线索设计案例,分解知识点,有明确的目的和要求,针对性强。
- (2) 选择有代表性的案例,突出重点知识的掌握和应用。
- (3) 将技术指导、代码与分析、应用提高、相关知识有机结合起来。
- (4) 注意新方法、新技术的应用。
- (5) 处理好具体实例与思想方法的关系、局部知识应用与综合应用的关系。
- (6) 强调实用性,培养应用能力。

本书中每一个案例的结构模式为“案例说明→案例目的→技术要点→代码及分析→应用扩展→相关知识及注意事项”。每一章均包含多个案例,并配有相应的习题。通过强化案例和实训教学,可以加深学生对理论知识的理解。

本书内容符合教育部高职高专规划教材指导思想与原则的要求,充分考虑了高职高专学生的培养目标和教学特点。在内容的组织上,本着由浅入深、循序渐进的原则,注重基本知识和基本概念的介绍,结合案例重点介绍实用性较强的内容,对难度过大的内容只作少量介绍,使学生有重点地掌握所学内容。

本书注重培养学生的实际应用技能和综合解决问题的能力,使学生能熟练掌握和运用理论知识解决实际问题,达到学以致用目的。

本书建议课程安排 84 课时,其中理论教学为 64 课时,实验教学为 20 课时。

建议任课老师在教学过程中,采用启发式案例教学的方法,注重培养学生的创新思维能力。

本课程的前导课程包括“C 语言程序设计”和“高等数学”,后继课程是“算法与数据结构”。

本书由潍坊学院的徐翠霞担任主编。潍坊学院的崔玲玲和山西大学工程学院的邵回祖担任副主编。其中,第 1 章由邵回祖编写;第 2 章和第 6 章由崔玲玲编写;第 3 章由邢台职业技术学院的杨平编写;第 4 章和第 5 章由徐翠霞编写;第 7 章由淄博职业学院的周彬和广东机电职业技术学院的陈翠松编写;第 8 章由陕西航空职业技术学院的欧林梅编写。徐翠霞负责全书审阅。

限于作者水平,书中难免会有疏漏之处,恳请读者批评指正,以使本书得以改进和完善。

编 者

2009 年 1 月

目 录

第 1 章 线性表	1
1.1 “顺序表的就地逆置”案例	2
1.1.1 案例实现过程	2
1.1.2 应用扩展	3
1.1.3 相关知识及注意事项	4
1.2 “一元多项式的加法运算”案例	14
1.2.1 案例实现过程	14
1.2.2 应用扩展	18
1.2.3 相关知识及注意事项	20
本章小结	32
习题 1	33
第 2 章 栈和队列	37
2.1 “后缀表达式的求值”案例	38
2.1.1 案例实现过程	38
2.1.2 应用扩展	39
2.1.3 相关知识及注意事项	41
2.2 “背包问题”案例	49
2.2.1 案例实现过程	49
2.2.2 应用扩展	52
2.2.3 相关知识及注意事项	54
2.3 “农夫过河问题”案例	55
2.3.1 案例实现过程	55
2.3.2 应用扩展	59
2.3.3 相关知识及注意事项	61
本章小结	72
习题 2	73
第 3 章 串、数组和广义表	77
3.1 “文学研究助手”案例	78
3.1.1 案例实现过程	78
3.1.2 应用扩展	80
3.1.3 相关知识及注意事项	81

3.2 “稀疏矩阵的转置”案例	88
3.2.1 案例实现过程	88
3.2.2 应用扩展	93
3.2.3 相关知识及注意事项	94
本章小结	103
习题 3	103
第 4 章 树和二叉树	107
4.1 “二叉树遍历的演示”案例	108
4.1.1 案例实现过程	108
4.1.2 应用扩展	110
4.1.3 相关知识及注意事项	110
4.2 “哈夫曼编码器”案例	128
4.2.1 案例实现过程	128
4.2.2 应用扩展	130
4.2.3 相关知识及注意事项	132
本章小结	141
习题 4	141
第 5 章 图	146
5.1 “无向图的连通性判断”案例	147
5.1.1 案例实现过程	147
5.1.2 应用扩展	150
5.1.3 相关知识及注意事项	150
5.2 “最小代价通信网”案例	162
5.2.1 案例实现过程	162
5.2.2 应用扩展	165
5.2.3 相关知识及注意事项	165
5.3 “教学计划的编制”案例	175
5.3.1 案例实现过程	175
5.3.2 应用扩展	178
5.3.3 相关知识及注意事项	178
本章小结	184
习题 5	184
第 6 章 查找	188
6.1 “分块查找”案例	189
6.1.1 案例实现过程	189

6.1.2 应用扩展	192
6.1.3 相关知识及注意事项	193
6.2 “利用二叉排序树实现单词统计”案例	200
6.2.1 案例实现过程	200
6.2.2 应用扩展	202
6.2.3 相关知识及注意事项	202
6.3 “散列表的双散列探测查找”案例	213
6.3.1 案例实现过程	213
6.3.2 应用扩展	216
6.3.3 相关知识及注意事项	216
本章小结	223
习题 6	223
第 7 章 排序	228
7.1 “双向起泡排序”案例	229
7.1.1 案例实现过程	229
7.1.2 应用扩展	230
7.1.3 相关知识及注意事项	232
7.2 “堆排序”案例	240
7.2.1 案例实现过程	240
7.2.2 应用扩展	242
7.2.3 相关知识及注意事项	243
7.3 “基数排序”案例	250
7.3.1 案例实现过程	250
7.3.2 应用扩展	252
7.3.3 相关知识及注意事项	253
本章小结	256
习题 7	257
第 8 章 综合实训	262
实训 1 停车场管理	263
实训 2 五叉路口交通灯的管理系统	270
本章小结	276
习题 8	276
参考文献	277

第1章

线 性 表

教学目标与要求：本章主要介绍线性表的逻辑结构和各种存储表示方法，以及定义在逻辑结构上的各种基本运算和在存储结构上如何实现这些基本运算。通过本章学习，要求：

- 了解线性表中数据元素之间的逻辑关系以及在计算机中表示这种关系的两种不同的存储结构，即顺序存储结构和链式存储结构。
- 熟练掌握这两种存储结构的描述方法，如链表中的头结点、头指针和首元结点、尾元结点的区别及循环链表、双向链表的特点等。
- 熟练掌握在顺序存储结构上实现线性表基本运算的方法，特别是插入、删除与定位等运算。
- 熟练掌握在各种链表上实现线性表基本运算的方法，能在实际应用中选用适当的链表结构。
- 能够从时间和空间复杂度的角度比较线性表两种存储结构的不同特点及使用场合。

教学重点与难点：本章重点是熟练掌握用顺序表和单链表实现的各种基本算法及相关的时间性能分析，难点是能够使用本章所学到的基本知识设计有效算法解决与线性表相关的应用问题。

1.1 “顺序表的就地逆置”案例

1.1.1 案例实现过程

【案例说明】

顺序表的就地逆置就是利用原表的存储空间将线性表($a_1, a_2, \dots, a_n$)逆置为($a_n, a_{n-1}, \dots, a_1$), 并且此处要求只使用一个元素的辅助空间。程序运行结果如图 1.1 所示。

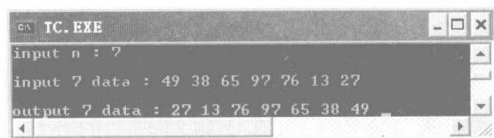


图 1.1 顺序表的就地逆置

【案例目的】

- (1) 掌握线性表的顺序存储结构。
- (2) 掌握顺序表的基本运算并能灵活应用。

【技术要点】

- (1) 空表或长度为 1 的表, 不做任何处理。
- (2) 表长大于或等于 2 时, 做如下处理。设两个指针 i 和 j , 分别指向顺序表的第 1 个元素和尾元素。若 $i < j$, 首先交换 $a[i]$ 和 $a[j]$, 然后修改指针, $i++$ 和 $j--$, 使它们分别指向顺序表的第 2 个元素和倒数第 2 个元素, 依次类推, 直到 $i \geq j$ 为止。

因为基本操作是对应元素的交换运算, 假设顺序表的长度为 n , 则需要进行 $n/2$ 次交换, 所以时间复杂度为 $O(n)$ 。

【代码及分析】

```
#define MAXSIZE 100                /*顺序表的最大长度*/
typedef int ElemType;              /*ElemType 为顺序表元素的类型*/
typedef struct {
    ElemType data[MAXSIZE];
    int length;                    /*顺序表的实际长度*/
} SqList;
void Reverse_SqList(SqList *L) {
    int i, j, n, t;
    n = L->length;
    if (n == 0 || n == 1) return;
    i = 0;
    j = n - 1;
    while (i < j) {
        t = L->data[i];
```

```

        L->data[i]=L->data[j];
        L->data[j]=t;
        i++;
        j--;
    }
}

void Creat_SqList(SqList *L,int n){
    int i;
    L->length=n;
    i=0;
    printf("\ninput %d data : ",n);
    while(i<n){
        scanf("%d",&L->data[i]);
        i++;
    }
}

void Print_SqList(SqList *L){
    int i,n;
    n=L->length;
    i=0;
    printf("\noutput %d data : ",n);
    while(i<n){
        printf("%d ",L->data[i]);
        i++;
    }
}

main(){
    SqList L;
    int n;
    printf("\ninput n : ");
    scanf("%d",&n);
    Creat_SqList(&L,n);
    Reverse_SqList(&L);
    Print_SqList(&L);
}

```

1.1.2 应用扩展

已知顺序表中的元素以值递增的顺序排列，删除顺序表中所有值相同的多余元素，使操作后顺序表中所有元素的值均不相同。

```

int Del_SqList(SqList *L){
    /*删除有序顺序表中所有值相同的多余元素*/
    int i,j,k;
    int n=L->length;
    int count=0;
    ElemType t;
    for(i=0;i<n;i++){
        t=L->data[i];

```

```

        for(j=i+1;j<n;j++){
            if(L->data[j]==t) count++;
            else break;
        }
        if(count!=0){
            for(k=j;k<n;k++)/*删除与当前元素重复的 count 个多余元素*/
                L->data[k-count]= L->data[k];
            n=n-count;      /*元素个数减 count*/
            count=0;        /* count 清 0,为删除下一个元素的重复元素做准备*/
        }
    }
    L->length=n;
}

```

1.1.3 相关知识及注意事项

1. 数据结构的基本概念

1) 数据

数据(Data)是客观事物的数字、字符以及所有能输入计算机中并被计算机程序处理的符号的集合,它是计算机程序加工的“原料”。常见的数据可分为数值数据和非数值数据。数值数据包括整数、实数,主要用于工程计算、科学计算和商务处理等;非数值数据包括字符、文字、图形、图像、语音等。

2) 数据元素

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。一个数据元素可以由一个或多个数据项(Data Item)组成,其中能够唯一标识一个数据元素的数据项称为**关键字**。数据项是数据不可分割的最小单位。

例如,学籍管理系统中学生信息表的每一个学生记录就是一个数据元素,它包括学生的学号、姓名、性别、籍贯、出生年月、成绩等数据项。通常,在解决实际问题时是把每个学生的记录当作一个基本单位进行访问和处理的。

3) 数据结构

数据结构(Data Structure)是指按照某种关系组织起来的一组数据,按照一定的存储表示方式把它存储在计算机中,并在这些数据上定义了一个运算的集合。数据结构一般包括以下 3 个方面的内容:数据的逻辑结构、数据的存储结构以及数据的运算。

(1) 数据的逻辑结构。

对数据之间关系的描述称为数据的逻辑结构。在任何问题中,数据元素之间都不会是孤立的,在它们之间都存在着这样或那样的关系,根据数据元素之间关系的不同特性,通常有下列 4 类基本结构。

- ① 集合。在集合中,数据元素之间除了“同属于一个集合”的关系外,无其他关系。
- ② 线性结构。该结构中的数据元素之间存在着一对一的关系。
- ③ 树结构。该结构的数据元素之间存在着一对多的关系。
- ④ 图形结构或网状结构。该结构的数据元素之间存在着多对多的关系。

图 1.2 所示为上述 4 类基本结构的示意图。

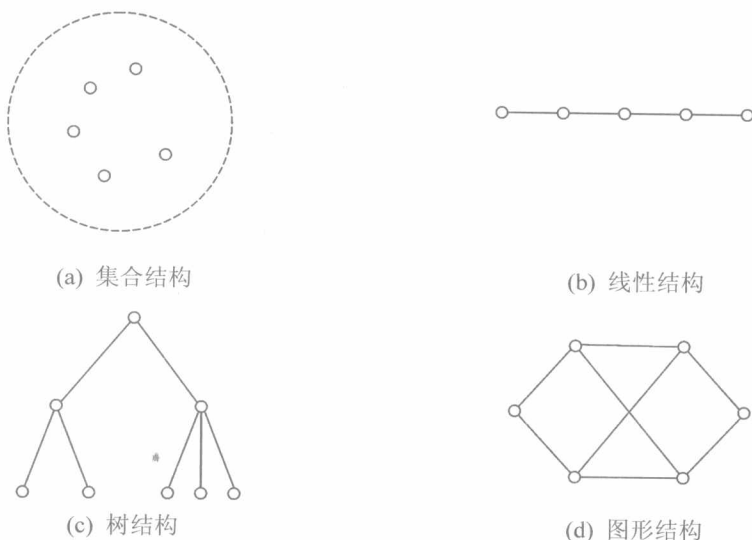


图 1.2 4 类基本结构

在形式上,数据的逻辑结构可以用一个二元组

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

来表示。其中, D 是数据元素的有限集, R 是 D 上关系的有限集。 R 中的每个关系都是从 D 到 D 的关系。

设 $B=(K,R)$ 是一个逻辑结构, r 是一个 K 到 K 的关系, $r \in R$ 。若 $k, k' \in K$, 且 $\langle k, k' \rangle \in r$, 则称 k' 是 k 的**后继**, k 是 k' 的**前驱**, 这时 k 和 k' 是相邻的结点或元素(相对关系 r 而言)。如果不存在一个 k' 使 $\langle k, k' \rangle \in r$, 则称 k 为关于 r 的**终端结点**(即无后继结点)。如果不存在一个 k' 使 $\langle k', k \rangle \in r$, 则称 k 为关于 r 的**开始结点**(即无前驱的结点)。如果 k 既不是终端结点, 也不是开始结点, 则称 k 是**内部结点**。

例如, 在计算机科学中, 复数可取如下定义。

$$\text{Complex} = (C, R)$$

其中, C 是含有两个实数的集合 $\{c_1, c_2\}$, $R=\{r\}$, 而 r 是一个 C 到 C 的关系。在关系 r 中, 只有一个有序对 $\langle c_1, c_2 \rangle$, 该有序对 $\langle c_1, c_2 \rangle$ 表示 c_1 是复数的实部、 c_2 是复数的虚部。

(2) 数据的存储结构。

数据的逻辑结构是从逻辑关系上来描述数据, 它与数据的存储无关, 是独立于计算机的。而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。

数据结构在计算机中的表示称为数据的**物理结构**, 也称为**存储结构**。它所研究的是数据结构在计算机中的实现方法, 包括数据结构中元素的表示及元素间关系的表示。

数据的存储结构可以采用顺序存储方法和链式存储方法。顺序存储方法是把逻辑上相邻的元素存储在物理位置相邻的存储单元中, 由此得到的存储表示称为顺序存储结构。顺

序存储结构是一种最基本的存储表示方法,通常借助于程序设计语言中的数组来实现。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设的指针来表示,由此得到的存储表示称为链式存储结构,链式存储结构通常借助于程序设计语言中的指针类型来实现。

数据的存储结构是逻辑结构在计算机中的实现,它是依赖于计算机的。

(3) 数据的运算。

数据的运算是在数据上所施加的一系列操作。数据的运算是定义在逻辑结构上的,而运算的具体实现要在存储结构上进行。数据的各种逻辑结构有各种相应的运算,每个逻辑结构都有一个运算的集合。例如,最常用的运算有:检索(查找)、插入、删除、更新、排序等。

2. 算法和算法评价

1) 算法及其特性

算法(Algorithm)是对特定问题求解步骤的一种描述。通俗地讲,一个算法就是一个解题方法。更严格地说,算法是由若干条指令组成的有限序列,其中的每一条指令表示一个或多个操作。一个算法应该具有下列特性。

(1) 有穷性。一个算法必须总是(对任何合法的输入值)在执行有穷步之后结束,且每一步都可在有穷时间内完成。

(2) 确定性。算法中的每一步必须有确切的含义,并且在任何条件下,算法只有唯一的一条执行路径,即对于相同的输入只能得出相同的输出。

(3) 可行性。一个算法是可行的,即算法中描述的运算都可以通过已经实现的基本运算的有限次执行得以实现。

(4) 输入。一个算法具有零个或多个输入,它们是在算法开始前赋给算法最初的量,这些输入取自特定的数据集合。

(5) 输出。一个算法具有一个或多个输出,这些输出是同输入之间存在某种特定关系的量。

算法的含义与程序十分相似,但又有区别。一方面,一个程序不一定满足有穷性。例如,对于操作系统,只要整个系统不遭破坏,它将永远不会停止,即使没有作业需要处理,它仍处于动态等待中。因此,操作系统不是一个算法。另一方面,程序中的指令必须是机器可执行的,而算法中的指令则无此限制。算法代表了对问题的解,而程序则是算法在计算机上的特定实现。一个算法若用程序设计语言来描述,则它就是一个程序。

2) 算法描述

算法可以使用各种不同的方法来描述,最简单的方法是使用自然语言。用自然语言来描述算法的优点是简单且便于人们对算法的阅读,缺点是不够严谨。

算法还可以使用程序流程图、N-S图等算法描述工具来描述,其特点是描述过程简洁、明了。使用以上两种方法描述的算法不能直接在计算机上执行,若要将它转换成可执行的程序,还有一个编程的问题。

算法也可以直接使用某种程序设计语言来描述,不过直接使用程序设计语言并不容易,