

21世纪计算机系列规划教材



数据结构

(C语言版) (第2版)



肖宏启 主 编
刘昌明 陈 锐 廖银花 副主编

魏怀明 主 审



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

21 世纪计算机系列规划教材

数据结构 (C 语言版)

(第 2 版)

肖宏启 主 编

刘昌明 陈 锐 廖银花 副主编

魏怀明 主 审

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书是《数据结构(C语言版)》的第2版,对常用的数据结构做了系统的介绍,力求概念清晰,注重实际应用。主要内容包括:数据结构的基本概念,算法描述和算法分析初步,线性表、链表、栈、队列、串、广义表、树、图等结构,查找和排序的各种方法,另外还用一章的篇幅详细介绍了链式存储结构以加深读者的理解。每章均列举了典型应用实例,并配有算法和程序以供教学和实践使用。

本书作为“21世纪计算机系列规划教材”之一,内容丰富,层次清晰,讲解深入浅出,可作为高等院校应用型本科、专科及高等职业院校计算机类专业学生数据结构课程的教材,也可以作为大学非计算机专业学生的选修课教材和计算机应用技术人员自学参考书。

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。

图书在版编目(CIP)数据

数据结构:C语言版 / 肖宏启主编. —2版. —北京:电子工业出版社,2014.1

21世纪计算机系列规划教材

ISBN 978-7-121-22041-8

I. ①数… II. ①肖… 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. ①TP311.12②TP312



中国版本图书馆CIP数据核字(2013)第284472号

策划编辑:程超群

责任编辑:程超群 特约编辑:张彬

印 刷:三河市鑫金马印装有限公司

装 订:三河市鑫金马印装有限公司

出版发行:电子工业出版社

北京市海淀区万寿路173信箱 邮编 100036

开 本:787×1092 1/16 印张:21 字数:538千字

印 次:2014年1月第1次印刷

印 数:3000册 定价:43.00元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010)88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010)88258888。

前 言

随着社会经济的高速发展,我国的高等教育已从精英教育走向大众化教育的发展阶段。国际高等教育的历史说明高等职业教育必将成为这一发展过程的主力军。对于我国这种从专科转型的高等职业教育,现今还处于探索阶段,教材建设是这一时期教育改革的重要方面。

高等职业教育是高等教育的一个新的类型,它与传统的普通高等教育既有紧密的联系,又有本质的区别。高等职业教育强调面向社会、生产、管理、服务第一线,培养技术应用型人才,学生毕业后,即可发挥其所学专长。所以,根据我国教育部的规定,高等职业教育需根据其自身的特点,建立自己的教材体系。

本教材第2版列入“21世纪计算机系列规划教材”,主要面向应用型本科及高等职业院校计算机专业的学生,教材内容的构造力求体现“以应用为主体”,强调理论知识的理解和运用,实现高校应用型本科及高等职业教育教学以实践体系、技术应用能力培养为主的目标,符合现代高等职业教育对教材的需求。

“数据结构”是计算机程序设计的重要理论基础,是计算机及其应用专业的一门重要基础课程和核心课程。它不仅是学习后继软件专业课程的先导,而且已成为其他工科类专业的热门选修课程。

本书共9章。第1章阐述数据、数据结构和算法等基本概念。第2~7章分别讨论线性表、链表、栈、队列、串、广义表、树和二叉树以及图等基本数据结构及其应用,其中,第3章专门总结了链式存储结构的基本概念和应用,为学好后面的各类数据结构打好扎实的基础。第8~9章讨论查找和排序的各种实现方法和实用分析。

第2版教材进行了大量实用例子的补充和调整,对大量抽象、难懂的概念进行深入浅出的分析和讲解。将第1版教材中每章的实验子系统变更为验证性上机实训,并在附录里增加了设计性上机实训题供教师和学生课程设计阶段选用。

长期以来,由于数据结构课程自身的抽象性和严密性,教师大都感觉数据结构课程难教,学生普遍反映数据结构课程难学,学生很难独立完成算法的实现。基于上述问题,我们在编写本教材时充分考虑了学生的知识结构和教师的教学方法。本教材的编写宗旨是,既注重原理又注重实践,既注重抽象思维又注重形象思维,既方便自学又方便教学。

本书的特点如下。

(1) 对基础理论知识的阐述由浅入深、通俗易懂。内容组织和编排以应用为主线,略去了一些理论推导和数学证明的过程,淡化算法的设计分析和复杂的时空分析。

(2) 各章都列举分析了很多实用的例子,这有助于学生加深对基础理论知识的理解和实际应用的能力培养。

(3) 考虑到此课程的先导课程是“C语言程序设计”,书中算法均采用可在计算机上运行的C语言程序来描述。这样,降低了算法设计的难度,使学生能更直观形象地理解这些算法。鉴于微软对于VC++6.0早已停止维护升级,对于出现的不兼容问题已不再解决,本书提供的实现代码均在简单易用的Dev-C++平台上编译通过,并给出了所有程序的运行结果。若



教师教学采用 TC 环境, 只需在相应实现代码中加上相关头文件即可 (Dev-C++ 平台安装包及使用说明书在教学资源包中下载)。

(4) 为配合本书的教学, 特编制了多媒体课件, 对学生加深理解基本概念起到更直观的效果。本书配套的多媒体课件、书中所有算法及上机实训源代码和习题答案可在华信教育资源网 (www.hxedu.com.cn) 下载, 或通过 E-mail 向编者索取: xiaohongqi2000@163.com。

(5) 在教材中使用 “▲思考” 标志, 提出问题拓展学生思维。在教学中可恰到好处地启发学生的思维。

(6) 为避免 C 语言中数组的第一个元素的下标为 0 给学习和讲授带来的不便, 本书在没有特别声明的地方均不使用 C 语言中数组下标为 0 的元素。

本书由肖宏启整体构思、统稿和修改, 在多位教师长期从事数据结构课程教学的经验基础上, 经多次反复磋商和共同讨论定稿, 是多位作者共同合作的产物。具体分工如下: 第 1、3、8、9 章由贵州航天职业技术学院肖宏启编写; 第 2、4 章由中国人民解放军国防大学陈锐编写; 第 6 章由江西渝州科技职业学院廖银花编写; 第 7 章及附录部分由重庆电子工程职业学院刘昌明编写; 魏怀明副教授编写了第 5 章并详细审阅了全书; 陈美成副教授、韦军博士审阅了该书并提供了许多宝贵的意见, 陈元春副教授提供了本书的大部分习题。

本书编写过程中参考了许多作者的大量文献资料和国内外优秀教材, 电子工业出版社对本书的出版给予了大力支持和帮助, 作者谨此一并致以诚挚的谢意。

本教材讲课时数可为 60~72 学时, 上机时数可灵活安排。教师可根据学时数、专业和学生的实际情况, 选讲应用举例及分析中一些较难的例子。

由于编写教材的时间紧张, 难免存在疏漏, 敬请读者批评指正。

编 者
2013 年 8 月

目 录

第1章 绪论.....	1
1.1 什么是数据结构.....	2
1.1.1 数据结构的起源.....	2
1.1.2 数据结构研究的内容.....	4
1.1.3 学习数据结构的必要性.....	5
1.1.4 如何学好数据结构.....	6
1.2 数据的逻辑结构.....	6
1.2.1 基本概念.....	6
1.2.2 逻辑结构的描述.....	7
1.3 数据的存储结构.....	8
1.4 算法和算法效率分析.....	9
1.4.1 算法.....	9
1.4.2 影响算法效率的因素.....	10
1.4.3 算法效率的评价.....	11
上机实训.....	13
习题.....	14
第2章 线性表.....	18
2.1 内存静态分配.....	19
2.2 线性表的定义与基本操作.....	19
2.2.1 线性表的定义.....	19
2.2.2 线性表的基本操作.....	20
2.3 线性表的顺序存储结构.....	21
2.3.1 顺序表.....	21
2.3.2 顺序表上基本运算的实现.....	22
2.4 应用举例及分析.....	28
上机实训.....	30
习题.....	31
第3章 链表.....	32
3.1 内存动态分配.....	33
3.1.1 函数 malloc().....	33
3.1.2 函数 free().....	37
3.2 线性表的链式存储.....	38
3.2.1 线性链表.....	38
3.2.2 单链表上的基本运算.....	39



3.2.3	循环链表	56
3.2.4	双向链表	57
3.3	顺序表和链表的比较	59
3.4	应用举例及分析	60
	上机实训	67
	习题	68
第 4 章	栈和队列	70
4.1	栈	71
4.1.1	栈的定义及基本操作	71
4.1.2	栈的顺序存储结构	72
4.1.3	栈的链式存储结构	79
4.2	队列	83
4.2.1	队列的定义及基本操作	83
4.2.2	队列的顺序存储结构	84
4.2.3	队列的链式存储结构	93
4.3	应用举例及分析	99
	上机实训	106
	习题	110
第 5 章	串和广义表	114
5.1	串的定义和基本运算	115
5.1.1	串的定义	115
5.1.2	串的基本运算	115
5.2	串的实现	116
5.2.1	定长顺序存储	116
5.2.2	链式存储	117
5.2.3	堆分配存储	118
5.3	串基本运算的实现	120
5.4	广义表	133
5.4.1	广义表的定义和性质	133
5.4.2	广义表的存储	134
5.5	应用举例及分析	136
	上机实训	139
	习题	140
第 6 章	树和二叉树	143
6.1	树	144
6.1.1	树的定义	144
6.1.2	树的基本术语	145
6.1.3	树的存储	146
6.2	二叉树	150

6.2.1	二叉树的定义	150
6.2.2	二叉树的性质	151
6.2.3	二叉树的存储	153
6.2.4	二叉树的建立	156
6.3	遍历和恢复二叉树	157
6.3.1	遍历二叉树	157
6.3.2	恢复二叉树	163
6.4	树、森林与二叉树的转换	165
6.4.1	一般树转换为二叉树	165
6.4.2	森林转换为二叉树	167
6.4.3	二叉树转换为树和森林	167
6.5	二叉树的应用举例	168
6.6	哈夫曼树及其应用	171
6.6.1	哈夫曼树的引入	171
6.6.2	哈夫曼树的建立	174
6.6.3	哈夫曼编码	175
	上机实训	180
	习题	181
第 7 章	图	186
7.1	图的定义和术语	187
7.1.1	图的定义	187
7.1.2	图的相关术语	188
7.1.3	图的基本操作	191
7.2	图的存储表示	191
7.2.1	邻接矩阵	191
7.2.2	邻接表	194
7.2.3	边集数组	198
7.3	图的遍历	199
7.3.1	深度优先搜索	199
7.3.2	广度优先搜索	205
7.4	生成树和最小生成树	211
7.4.1	生成树和生成森林	211
7.4.2	最小生成树	213
7.4.3	普里姆算法	213
7.4.4	克鲁斯卡尔算法	217
7.5	单源最短路径	222
7.5.1	单源最短路径的概念	222
7.5.2	求单源最短路径的方法	222
7.6	AOV 网与拓扑排序、拓扑排序的实现	227



7.6.1	AOV 网与拓扑排序.....	227
7.6.2	拓扑排序的实现.....	228
7.7	AOE 网与关键路径、关键路径的确定.....	232
7.7.1	AOE 网与关键路径.....	232
7.7.2	关键路径的确定.....	233
	上机实训.....	233
	习题.....	236
第 8 章	查找.....	239
8.1	查找的基本概念.....	240
8.2	静态查找表.....	241
8.2.1	顺序查找.....	242
8.2.2	二分查找.....	244
8.2.3	索引顺序查找.....	249
8.3	动态查找表.....	250
8.3.1	二叉排序树定义.....	251
8.3.2	二叉排序树的插入和生成.....	251
8.3.3	二叉排序树的删除.....	254
8.3.4	二叉排序树上的查找.....	256
8.4	散列表.....	263
8.4.1	散列表与散列函数.....	263
8.4.2	散列函数的构造方法.....	264
8.4.3	处理冲突的方法.....	266
8.4.4	散列表的查找及分析.....	268
	上机实训.....	272
	习题.....	274
第 9 章	排序.....	278
9.1	基本概念.....	279
9.2	插入排序.....	280
9.2.1	直接插入排序.....	280
9.2.2	二分插入排序.....	284
9.2.3	希尔排序.....	286
9.3	交换排序.....	289
9.3.1	冒泡排序.....	289
9.3.2	快速排序.....	294
9.4	选择排序.....	298
9.4.1	简单选择排序.....	298
9.4.2	堆排序.....	300
9.5	归并排序.....	306
9.6	各种排序方法的比较.....	309

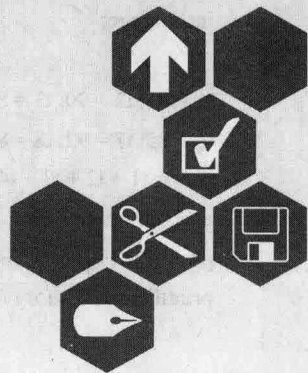
上机实训.....	311
习题.....	312
附录 A 设计性上机实训	316
课程设计指导	317
课程设计参考题目	317
课程设计（实训）报告参考格式	321
参考文献.....	325

1

绪 论

本章内容概要:

计算机科学是一门研究数据表示和数据处理的科学。计算机在发展初期，其应用范围是数值计算，所处理的数据都是整型、实型、布尔型等简单数据，并以此为加工对象进行数值型程序的设计。后来，随着电子技术的发展，计算机逐渐进入商业、制造业等其他领域，从而广泛地应用于数据处理的和过程控制。与此相应，计算机处理的数据也不再是简单的数值，而是字符串、图像、图形、声音、视频等复杂的数据。数据是计算机化的信息，是计算机可以直接处理的最基本和最重要的对象。在计算机应用领域中进行科学计算或数据处理、过程控制、对文件的存储和检索及数据库技术等，都是对数据进行加工处理的过程。因此，要设计出一个结构好、效率高的程序，必须研究数据的特性及数据间的相互关系及其对应的存储表示，而数据结构就是一门研究这些问题的课程，并利用这些特性和关系设计出相应的算法和程序。本章将介绍数据结构和算法分析的基本概念。





1.1 什么是数据结构

1.1.1 数据结构的起源

首先,我们用两个简单的 C 语言程序范例来说明数据结构与程序设计的关联性,表 1.1 所示是 10 次 C 语言课程的测验成绩,请读者先思考一下,然后设计一个小程序计算这 10 次测验的总分和平均分。

表 1.1 测验成绩

测 验	成 绩
1	81
2	90
3	80
4	59
5	70
6	85
7	92
8	84
9	100
10	78

【程序 1-1】

```
/* ===== */
/* 程序实例: 1-1.c */
/* 计算总分和平均分 */
/* ===== */

void main()
{
    int t1,t2,t3,t4,t5;           /*各次的成绩*/
    int t6,t7,t8,t9,t10;
    int sum;                      /*总分*/
    int average;                 /*平均分*/

    t1 = 81; t2 = 90; t3 = 80; t4 = 59; t5 = 70;
    t6 = 85; t7 = 92; t8 = 84; t9 = 100; t10 = 78;
    sum = t1 + t2 + t3 + t4 + t5 + t6 + t7 + t8 + t9 + t10; /*计算总分*/
    average = sum / 10;          /*计算平均分*/
    printf("输出总分: %d\n",sum); /*输出总分*/
    printf("输出平均分: %d\n",average); /*输出平均分*/
}
```

程序运行结果:

输出总分: 819

输出平均分: 81

这是一个很简单的 C 语言程序, 相信并不需要多费笔墨来解释。如果读者想到的方法是程序 1-1, 必须很抱歉地建议您, 需要先加强一下 C 语言的程序设计功力。不过也不必因此而感到难过或自责, 毕竟大多数的读者, 包括笔者在内, 都一样曾经历过这么一段学习程序设计的过程。

程序 1-1 使用数个内存变量存储考试成绩。这种方法的扩充性不是很好, 因为如果考试的次数改变了, 增加成为 15 次或减少成为 8 次, 整个程序就需要很大的修改。事实上, 从测验次数和成绩的关系可得知, 用数组结构保存测验的成绩是一种更好的方法。程序 1-2 正是使用的这种方法。

【程序 1-2】

```
/* ===== */
/* 程序实例: 1-2.c */
/* 计算总分和平均分 */
/* ===== */

void main()
{
    int t[10] = { 81,90,80,59,70,85,92,84,100,78 };
    int sum; /*总分*/
    int average; /*平均分*/
    int i;

    sum = 0; /*设置总分初值*/
    for ( i = 0; i < 10; i++ )
        sum += t[i]; /*计算总分*/
    average = sum / 10; /*计算平均分*/
    printf("输出总分: %d\n",sum); /*输出总分*/
    printf("输出平均分: %d\n",average); /*输出平均分*/
}
```

程序运行结果:

输出总分: 819

输出平均分: 81

相信读者在比较程序 1-1 和程序 1-2 后, 一定可以发现程序 1-2 使用的方法比较好。可是为什么读者会认为程序 1-2 比较好呢? 是经验吗? 事实上, 这是因为程序 1-2 使用了比较好的数据结构解决问题。虽然这两个程序都可以正确地解决问题, 但是采用不同的方式保存成绩数据, 进而形成不同的程序设计方式。所以如何选择最佳的数据结构来解决程序问题, 是需要学习数据结构的重要原因。



数据结构这门课程的起源是程序设计的经验累积，自从计算机（电脑）发明以来，科技一日千里，计算机知识随处可得，目前的一般读者根本无法理解到早期程序设计师的辛酸。程序设计对于早期的程序设计师来说，是一种艺术而不是技术。因为参考的数据难得，每一位程序设计师在追求计算机知识的过程中，都曾经经历过一段非常艰苦的时光。逐渐地，由这些先辈留下来的宝贵的程序设计经验，确实能够有效地解决一些程序问题，而这些方法便成为了一门学问。这就是“数据结构（Data Structure）”这门课程的起源。

1.1.2 数据结构研究的内容

早期人们都把计算机理解为数值计算工具，使用计算机的目的主要是处理数值计算问题。当我们使用计算机来解决一个具体问题时，一般需要经过下列几个步骤：首先要根据该具体问题抽象出一个适当的数学模型，然后设计或选择一个解此数学模型的算法，最后编制、运行并调试程序，直到实际问题被解决。

由于当时所涉及的运算对象是简单的整型、实型或布尔类型数据，所以程序设计者的主要精力集中于程序设计的技巧上，而无须重视数据结构。随着计算机软、硬件的发展和应用领域的不断扩大，计算机处理的对象更多地转向非数值计算问题，这类问题涉及的数据结构更为复杂，数据元素之间的相互关系一般无法用数学方程式加以描述，此时解决这类问题的关键不再是数学分析和计算方法，而是必须建立相应的数据结构来进行描述，分析问题中所用到的数据是如何组织的，研究数据之间的关系如何，进而设计出合适的数据结构，有效地解决问题。下面所列举的就是属于这一类的具体问题，用以说明什么是数据结构和数据结构所研究的内容。

例1.1 某班级学生的学籍档案信息如表1.2所示。

表 1.2 某班级学生的学籍档案信息表

学 号	姓 名	性 别	年 龄	入 学 成 绩
2013001	张 三	男	21	80
2013002	刘薇薇	女	23	90
2013003	马 琳	男	22	78
⋮	⋮	⋮	⋮	⋮
2013050	李 明	男	22	85

这个学籍档案信息表中共有 50 个学生。我们可以把表中每个学生的信息看成一条记录并称之为一个节点，表中的每个节点由 5 个数据项组成。该学籍档案信息表由 50 个节点组成，每个节点排列有先后次序，形成一种线性关系。这是一种典型的数据结构，我们称这种数据结构为线性表。

对该表的主要操作有：在给出学号时，如何在表中快速查找到所对应的学生的信息；若有学生退学，如何删除该学生的记录；若有新生加入班级，如何在表中插入一条新记录。这些都是数据结构要研究的内容。

例 1.2 航天学院教学行政机构示意图如图 1.1 所示。

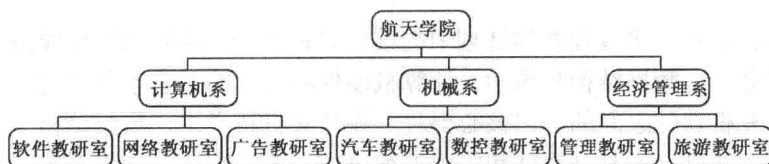


图 1.1 航天学院教学行政机构示意图

对于航天学院的教学行政机构，可以把该学院的名称看成树根，把下设的若干系看成它的树枝中间节点，把每个系的教研室看成树叶。树中的每个节点可以包含较多的信息，节点之间的关系不再是顺序的，而是分层、分叉的一对多的非线性结构。这也是我们常用的一种数据结构，我们称之为树形结构。树形结构的主要操作有遍历、查找、插入和删除等。

例 1.3 城市之间建立通信网络的问题，如图 1.2 所示。

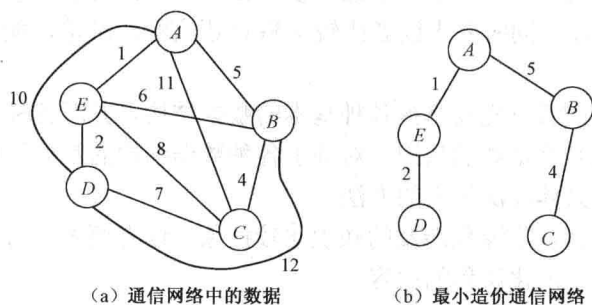


图 1.2 用图描述通信网络问题

在 n 个城市之间建立通信网络，要求在其中任意两个城市之间都有直接或间接的通信线路；在已知某些城市之间直接通信线路造价的情况下，使网络造价最低。

当 n 很大时，这样的问题只能用计算机来求解。我们用图 1.2 (a) 中描述的关系来说明：图中的每个小圆圈表示一个城市，两个圆圈之间的连线表示对应城市之间的通信线路，连线上的数值表示该通信线路的造价。这一描述结构为图状结构，利用计算机可以求出满足要求的最小造价通信网络，如图 1.2 (b) 所示。如图 1.2 (a) 所示的这种数据结构中，数据之间的关系是多对多的非线性关系，我们称这种数据结构为图形结构。

由以上三个例子可见，描述这类非数值计算问题的数学模型不再是数学方程，而是诸如表、树、图之类的数据结构。因此，可以说数据结构课程是主要研究非数值计算的程序设计问题中所出现的计算机操作对象以及它们之间的关系和运算的学科。

学习数据结构的目的是了解计算机处理对象的特性，将实际问题中所涉及的处理对象在计算机中表示出来并对它们进行处理。与此同时，通过算法训练来提高我们的思维能力，通过程序设计的技能训练来促进综合应用能力和专业素质的提高。

1.1.3 学习数据结构的必要性

随着计算机运行速度的加快和存储（内存）容量的不断加大，有的人可能认为程序的运行效率变得越来越不重要了。然而，计算机的功能越强大，人们就越想去尝试解决更加复杂的问题。而更复杂的问题需要更大的计算量，这使得对高效率的程序的需求更加明显，工作越复杂就越偏离人们的日常经验。因此，学习数据结构，掌握各种前人设计的算法的运行效率并自己设计高效率的算法是非常必要的。



数据结构不仅是计算机专业教学计划中的核心课程之一,而且已逐步成为非计算机专业的主要选修课程之一。数据结构与数学、计算机硬件和软件的关系十分密切,是介于数学、计算机硬件和计算机软件之间的一门核心课程。在计算机科学中,数据结构不仅是一般非数值计算程序设计的基础,而且是设计和实现汇编语言、编译程序、操作系统、数据库系统以及其他系统程序和大型应用程序的重要基础。打好数据结构课程的扎实基础,对于学习计算机专业其他课程,如编译原理、数据库系统原理、计算机网络基础等都十分有益。

1.1.4 如何学好数据结构

本课程(以本书为教材)的先导课程是“C 语言程序设计”。要想学好本课程,首先要求读者已经掌握了基本的 C 语言知识,掌握了模块化程序设计的基本思想,能够利用 C 语言熟练编写一些简单的程序。同时要求读者比较熟悉 C 语言编程环境,能够熟练地编辑、调试及运行 C 语言程序。

在学习本课程时,读者首先要掌握各种基本的数据结构,并对各种数据结构的逻辑特性和物理特性(存储结构)有足够的认识。对基于各种数据结构的常见操作及其算法要重点掌握,并要了解评价某个具体算法优劣的方法。

相对于其他课程而言,本课程涉及的知识比较抽象。读者要多思考、多做练习题、多上机实践,才能真正理解、消化课程的内容。

1.2 数据的逻辑结构

1.2.1 基本概念

1. 数据

数据(Data)是信息的载体,是描述客观事物的符号,是能够被计算机识别、存储和加工处理的符号集合。计算机科学中,数据的含义相当广泛,是指能被计算机加工处理的所有对象,它可以是数值数据,也可以是非数值数据。数值数据是一些整数、实数或复数等数值类型,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。

我们所说的数据,其实就是符号,但这些符号必须具备两个前提:可以输入到计算机中;能被计算机程序处理。

对于整型、实型等数值类型,可以进行数值计算。对于字符数据类型,就需要进行非数值处理。而声音、图像、视频等其实是通过编码的手段变成字符数据来处理的。

2. 数据元素

数据元素(Data Element)是数据的基本单位,在计算机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若干数据项组成,也可以只由一个数据项组成,例如,表 1.2 “某班级学生的学籍档案信息表”中的一条记录、图 1.2 “用图描述通信网络问题”中的一个城市都可称为一个数据元素。表 1.2 中每个学生的学籍信息作为一个数据元素,在表中占一行,每个数据元素由学号、姓名、性别、年龄和入学成绩 5 个数据项组成。数据元素又被称为元素、节点(node)、顶点、记录(record)等。

3. 数据项

数据项 (Data Item) 是数据不可分割的、具有独立意义的最小数据单位, 是对数据元素属性的描述。比如人这样的数据元素, 可以有眼、耳、鼻、嘴、手、脚这些数据项, 但也可以有姓名、年龄、出生地址、联系电话等数据项, 具体包括哪些数据项, 要根据我们做的系统决定。数据项又称为域或字段 (Field)。在表 1.2 中, 每个数据元素由 5 个数据项组成。

4. 数据类型

数据类型 (Data Type) 是一组性质相同的值的集合以及定义在这个值的集合上的一组操作的总称。每个数据项属于某一确定的基本数据类型。如表 1.2 中, 学号为数值型、姓名为字符型等。

5. 数据对象

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, 整数数据对象是集合 $N=\{0, \pm 1, \pm 2, \dots\}$, 字母字符数据对象是集合 $C=\{'A', 'B', \dots, 'Z'\}$, 如表 1.2 所示的学籍表也可看做一个数据对象。由此可看出, 不论数据元素集合是无限集 (如整数集)、有限集 (如字符集), 还是由多个数据项组成的复合数据元素 (如表 1.2 “某班级学生的学籍档案信息表”), 只要性质相同, 就都是同一个数据对象。

6. 数据结构

数据结构 (Data Structure) 的基本含义是指数据元素之间的关系, 它是按照某种关系组织起来的一批数据, 以一定的存储方式把它们存储到计算机存储器中, 并在这些数据上定义了一个运算的集合。在任何问题中, 数据元素都不是孤立存在的, 而是在它们之间存在着某种关系, 数据元素之间的这种相互关系就称为结构, 带有结构的数据对象称为数据结构。

1.2.2 逻辑结构的描述

1. 数据逻辑结构的基本分类

根据数据元素间关系的不同特性, 数据的逻辑结构通常划分成下面四种基本结构。

(1) 集合。在集合结构中, 数据元素除了同属于一个集合外不存在任何关系, 这是数据结构的一种特殊情况。集合是元素关系极为松散的一种结构, 各个数据元素是“平等”的, 它们的共同属性是“同属于一个集合”。数据结构中的集合关系类似于数学中的集合, 不在本书的讨论范围之内。

(2) 线性结构。该数据结构中的数据元素之间存在着一对一的关系。

(3) 树形结构。该数据结构中的数据元素之间存在着一对多的关系。

(4) 图形结构。该数据结构中的数据元素之间存在着多对多的关系, 又称为网状结构。

上述四种基本结构如图 1.3 所示。

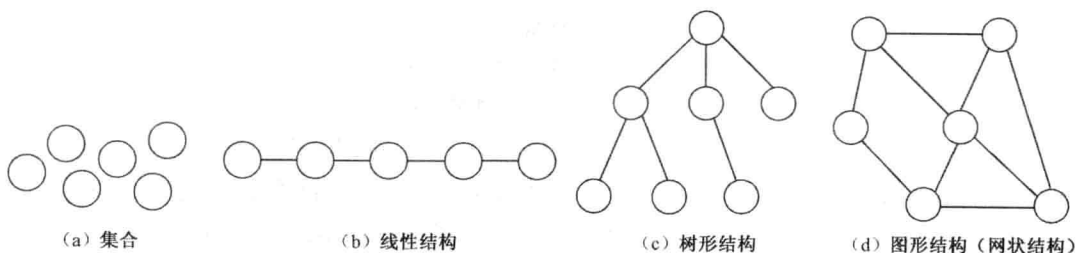


图 1.3 四种基本逻辑结构示意图