

高等院校信息技术规划教材

数据结构

闫玉宝 徐守坤 编著



清华大学出版社

TP311.12/163

2008

高等院校信息技术规划教材

数据结构

闫玉宝 徐守坤 编著

清华大学出版社
北京

内 容 简 介

本书围绕简明、易学、易用的原则来编写,在系统地介绍各种数据类型的数据结构的基础上,进行应用实例分析与 C 语言描述实现,示例程序很容易地利用 Visual C++ 或 TC 等开发环境实现,较全面地把数据结构的理论与实践结合起来。全书共 9 章,包括绪论、线性表、栈和队列、串、数组和广义表、树、图、查找、排序等内容。教材中除详细的应用实例外,各章又配备了适量的习题,以便于读者理解和掌握数据结构的基本理论和方法。

本书可作为各类大专院校的计算机科学与技术 and 信息类相关专业的教材或学习参考书。本书要求读者熟悉 C 或 C++ 语言。

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 13501256678 13801310933

图书在版编目(CIP)数据

数据结构/闫玉宝,徐守坤编著. —北京:清华大学出版社,2008.3

(高等院校信息技术规划教材)

ISBN 978-7-302-16849-2

I. 数… II. ①闫… ②徐… III. 数据结构 IV. TP311.12

中国版本图书馆 CIP 数据核字(2008)第 006820 号

责任编辑:袁勤勇 赵晓宁

责任校对:时翠兰

责任印制:何 芊

出版发行:清华大学出版社

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

c-service@tup.tsinghua.edu.cn

社 总 机:010-62770175

投稿咨询:010-62772015

地 址:北京清华大学学研大厦 A 座

邮 编:100084

邮购热线:010-62786544

客户服务:010-62776969

印 刷 者:北京密云胶印厂

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185×260 印 张:13.75

版 次:2008 年 3 月第 1 版

印 数:1~4000

定 价:19.80 元

字 数:312 千字

印 次:2008 年 3 月第 1 次印刷

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:010-62770177 转 3103 产品编号:022946-01

编委会名单

主 任： 朱 敏

副主任： 王正洪 鲁宇红 焦金生

成 员： (按拼音排序)

常晋义	邓 凯	范新南	高佳琴	高玉寰	龚运新
顾建业	顾金海	林 罡	刘训非	马正华	沈孟涛
唐 全	王继水	王 骏	王 晴	王志立	吴访升
肖 玉	杨长春	袁启昌	张旭翔	张 燕	赵明生
郑保平	郑成增	周凤石			

策划编辑： 袁勤勇

序

preface

在科教兴国方针的指引下,我国高等教育进入了一个新的历史发展时期,招生规模 and 在校生数量都有了大幅度的增长。我们在进行着世界上规模最大的高等教育。与此同时,对于高等教育的研究和认识也在不断深化。高等学校要明确自己的办学方向和办学特色,这既是不断提高高等教育水平的必然要求,更是高校不断发展和壮大必须首先考虑的问题。

教育部领导明确提出,高等教育应多元化,高等院校应实施分类分层次教学,这是高等教育大众化的必然结果,也是市场对人才需求的客观规律所致。因此要有相当部分的高等院校致力于培养应用型人才。此类院校在计算机教学中如何实现自己的培养目标,如何选择适用的应用型教材,已成为十分重要和迫切的任务。应用型人才的培养不能简单照搬研究型人才的培养模式,要在丰富的实践基础上认真总结,摸索新形势下的教学规律,在此基础上设计相关课程、改进教学方法,同时编写与之相适应的应用型教材。这一工作是非常艰巨的,也是非常有意义的。

在清华大学出版社的大力支持和配合下,于2003年成立了应用型教材编委会。编委会汇集了众多高等院校的实践经验,并经过集中讨论和专家评审,遴选了一批优秀教材,希望能够通过这套教材的出版和使用,促进应用型人才培养的实践发展,为建立新的人才培养模式作出贡献。

我们编写应用型教材的主要出发点是:

1. 适应新形势下教育部“质量工程”对高等教育的要求以及市场对应用型人才的需求。
2. 计算机科学技术和信息技术发展迅速,教材内容和教学方式应与之相适应,适时地进行更新和改进。
3. 教育技术的发展对教材建设提出了更高的要求,教材将呈现

出纸介质出版物、电子课件以及网络学习环境等相互配合的立体化形态。

4. 根据不同的专业要求,突出应用,使理论与实践更加紧密结合。

以此为目标,我们将努力编写一套全新的、有实用价值的应用型计算机教材。经过参编教师的努力,第一批教材已经面世。教材将滚动式地不断更新、修正、提高,逐渐树立起自己的品牌。希望使用本系列教材的广大师生能对我们的教材提出宝贵的意见,共同建设具有应用型特色的精品教材。

朱 敏

2006年5月

前言

foreword

“数据结构”是计算机科学与技术、通信和信息工程等专业的一门重要的专业技术基础课程。本教程较系统地介绍了软件设计中常用的数据结构以及相应的存储结构和实现算法;介绍了常用的多种查找和排序技术,并对其进行了性能分析和比较,内容非常丰富。在计算机科学中,数据结构不仅是一般程序设计(特别是非数值计算的程序设计)的基础,而且是设计和实现编译程序、操作系统、数据系统及其他系统程序和大型应用程序的重要基础,本课程的学习将为后续课程的学习以及软件设计水平的提高打下良好的基础。因此,数据结构课程是计算机专业的一门核心的关键性课程。

本教程的编写以“内容丰富,通俗易懂,简明实用”为宗旨。以基本知识点为基础,实际应用为重点,对各章节进行组织编写,通过典型的示例掌握各数据类型的理解 and 应用。每一章后面选配了适量的习题,加强对知识点的理解和实践。

本教程由闫玉宝、徐守坤编著,全文由闫玉宝统稿。参编人员有李宁、胡超,其中闫玉宝参与了第1、3、6章的编写,徐守坤参与了第2、5章的编写,李宁参与了第4、7章的编写,胡超参与了第8、9章的编写。

在编写本书过程中,参考了一些国内外优秀的相关教材,在此向这些教材的作者表示衷心感谢。

由于编者水平有限,错误之处在所难免,敬请同行及读者批评指正。

编 者

目录

Contents

第 1 章	绪论	1
1.1	数据结构的研究内容和作用	1
1.2	数据结构相关的基本概念	3
1.2.1	基本概念	3
1.2.2	数据结构	3
1.3	算法及其描述和分析	5
1.3.1	算法及其设计原则	5
1.3.2	算法的描述	5
1.3.3	算法效率的衡量方法	6
1.3.4	算法的存储空间需求	7
	习题 1	8
第 2 章	线性表	10
2.1	线性表的逻辑结构	10
2.1.1	线性表的定义	10
2.1.2	线性表的基本操作	11
2.2	线性表的顺序表示及其实现	12
2.2.1	顺序表	12
2.2.2	顺序表上的基本操作实现	13
2.2.3	顺序表应用举例	17
2.3	线性表的链式表示及其实现	18
2.3.1	单链表	18
2.3.2	单链表基本运算及其实现	19
2.3.3	循环链表	23
2.3.4	双向链表	24
2.3.5	静态链表	26
2.4	一元多项式的表示	27

2.5 线性表的应用举例	29
习题 2	30
第 3 章 栈和队列	33
3.1 栈	33
3.1.1 栈的定义及基本运算	33
3.1.2 栈的存储与操作实现	34
3.1.3 栈的应用举例	36
3.2 队列	43
3.2.1 队列的定义及基本运算	43
3.2.2 队列的存储表示和操作的实现	44
3.2.3 队列应用举例	47
习题 3	50
第 4 章 串	52
4.1 串的定义、表示和实现	52
4.1.1 串的基本概念	52
4.1.2 串的基本操作	53
4.2 串的存储和基本运算实现	54
4.2.1 串的定长顺序存储与基本运算实现	54
4.2.2 串的堆存储与基本运算实现	56
4.2.3 串的块链存储与基本运算实现	57
4.3 串的模式匹配	58
4.3.1 简单模式匹配算法	58
4.3.2 改进的模式匹配算法	59
4.4 串的应用举例	63
习题 4	65
第 5 章 数组和广义表	66
5.1 数组	66
5.1.1 数组的定义	66
5.1.2 数组的顺序存储结构	66
5.2 特殊矩阵的压缩存储	68
5.2.1 对称矩阵	68
5.2.2 下(上)三角矩阵	69
5.2.3 对角矩阵	69

5.3 稀疏矩阵的压缩存储	70
5.3.1 稀疏矩阵定义	70
5.3.2 稀疏矩阵的压缩存储方法——三元组表	70
5.3.3 稀疏矩阵的压缩存储方法——十字链存储法	74
5.4 广义表	75
5.4.1 广义表的概念和基本操作	75
5.4.2 广义表的存储结构	76
5.4.3 广义表基本操作的实现	79
习题 5	82
第 6 章 树和二叉树	84
6.1 树的类型定义	84
6.2 二叉树	85
6.2.1 二叉树的类型定义	85
6.2.2 二叉树的几个特性	86
6.2.3 二叉树的基本操作	87
6.3 二叉树的存储	87
6.3.1 顺序存储结构	87
6.3.2 链式存储结构	88
6.4 二叉树的操作	90
6.4.1 二叉树的遍历	90
6.4.2 二叉树的其他操作算法举例	92
6.5 线索二叉树	94
6.5.1 二叉树的线索链表	94
6.5.2 以中序线索链表为存储结构的中序遍历	96
6.5.3 线索链表的生成	96
6.6 树和森林	97
6.6.1 树的存储结构	97
6.6.2 树、森林与二叉树的相互转换	101
6.6.3 树、森林的遍历	102
6.6.4 应用举例	103
6.7 哈夫曼树及其应用	104
6.7.1 最优二叉树的定义	104
6.7.2 最优二叉树的构造方法	105
6.7.3 最优前缀编码	106
6.7.4 哈夫曼编码算法的实现	108
习题 6	110

第7章 图	112
7.1 图的基本概念	112
7.1.1 图的定义和术语	112
7.1.2 图的基本操作	116
7.2 图的存储结构	116
7.2.1 邻接矩阵	117
7.2.2 邻接表	119
7.3 图的遍历	121
7.3.1 深度优先遍历	121
7.3.2 广度优先遍历	123
7.4 图的连通性	124
7.4.1 图的连通性及其判定	124
7.4.2 生成树与最小生成树	125
7.5 有向无环图及其应用	129
7.5.1 有向无环图的概念	129
7.5.2 AOV网与拓扑排序	130
7.5.3 AOE网与关键路径	133
7.6 最短路径	139
7.6.1 求某一顶点到其余各顶点的最短路径	139
7.6.2 求任意一对顶点之间的最短路径	142
7.7 应用举例	144
习题7	148
第8章 查找	151
8.1 静态查找表	151
8.1.1 顺序查找	152
8.1.2 有序表的折半查找	153
8.1.3 分块查找	155
8.2 动态查找表	156
8.2.1 二叉排序树	156
8.2.2 平衡二叉树	160
8.2.3 B-树和B+树	165
8.3 哈希表	171
8.3.1 哈希表与哈希方法	171
8.3.2 常用的哈希函数	172
8.3.3 处理冲突的方法	174

8.3.4 哈希表的查找分析	176
习题 8	177
第 9 章 排序	179
9.1 内排序算法	179
9.1.1 插入排序	179
9.1.2 交换排序	186
9.1.3 选择排序	189
9.1.4 二路归并排序	193
9.1.5 基数排序	195
9.2 外排序算法	199
习题 9	200
参考文献	203

绪 论

自从发明计算机后,人类一直在探索人与计算机之间的沟通问题。从低级计算机程序语言到高级计算机程序语言,人类不断地提高人利用计算机解决现实社会中各类复杂问题的能力,这些语言符号我们称为“数据”,都能被计算机所理解并进行处理。计算机科学是一门研究数据表示和数据处理的科学,数据结构研究如何最有效地设计数据、处理数据的思想方法。

1.1 数据结构的研究内容和作用

计算机专家认为:算法+数据结构=程序设计,它说明了数据结构在程序设计中的作用。程序设计的实质即为计算机处理问题编制一组“指令”,首先需要解决两个问题:算法和数据结构。算法即处理问题的策略,而数据结构即为问题的描述模型。

很多数值计算问题的描述模型通常可用一组线性或非线性的代数方程组或微分方程组来描述,而大量非数值计算问题的描述模型正是本门课程要讨论的数据结构。

例 1.1 走迷宫问题。可用“穷解法”进行搜索,存在如何表示迷宫、搜索状态的问题,可由数组、栈表示,如图 1.1 所示。

例 1.2 九宫重排问题。由初始状态出发,依据一定的规则,最后到达目标状态。若要编制程序解决问题,首先要解决一个如何表示的问题。可由状态树表示。

在一个 3×3 的方格棋盘上放置 8 个标有 1、2、3、4、5、6、7、8 数字的将牌,留下一个空格(用 0 表示)。规定与空格上下左右相邻的将牌可以移入空格。问题要求寻找一条从某初始状态 S_0 到目标状态 S_g 的将牌移动路线。

开始结点 S_0 :

2	8	3
1	6	4
7	0	5

目标结点 S_g :

1	2	3
8	0	4
7	6	5

解路径：(操作序列：空格上移，空格上移，空格左移，空格下移，空格右移)

2	8	3
1	6	4
7	0	5

2	8	3
1	0	4
7	6	5

2	0	3
1	8	4
7	6	5

0	2	3
1	8	4
7	6	5

1	2	3
0	8	4
7	6	5

1	2	3
8	0	4
7	6	5

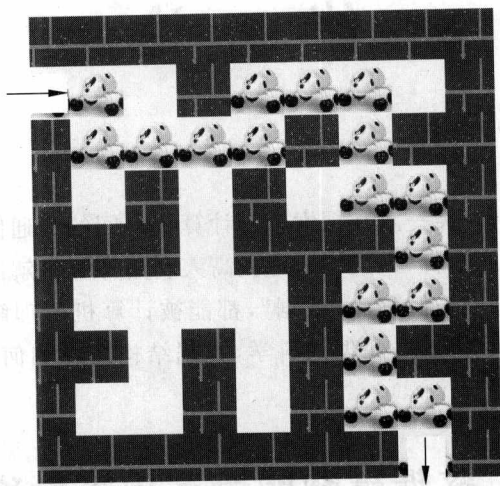
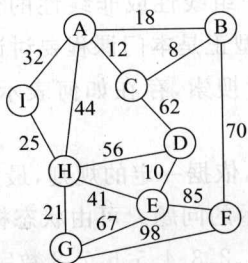
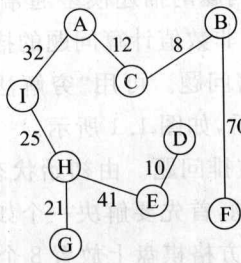


图 1.1 走迷宫

例 1.3 网线铺设问题。如图 1.2 所示，需为城市的各小区之间铺设网线，对 n 个小区只需铺设 $n-1$ 条管线，由于地理环境不同等因素使各条管线所需投资不同(如图 1.2 所标识)，如何使投资成本最低？这是一个讨论图的生成树的问题。



(a) 各小区间网线铺设成本示意图



(b) 实际网线铺设方案

图 1.2 网线铺设

以上所举例子中的数学模型正是数据结构要讨论的问题。因此，简单地说，数据结构是一门讨论“描述现实世界非数值计算问题的数学模型及其上的操作在计算机中如何表示和实现”的学科。

由数据结构研究的内容，决定了它在计算机科学与技术专业的基础核心地位。它也是操作系统、编译原理、数据库管理系统、软件工程和人工智能等专业课的先修课程。

1.2 数据结构相关的基本概念

1.2.1 基本概念

数据是所有能输入计算机,能由计算机程序处理的符号的总和。它包括所有的数字和字符。图形和声音等信息最后也都可以转化为“字符”进行处理,这些字符和数字是客观信息的一种描述。

数据元素是数据的基本单位,在程序中通常作为一个整体进行考虑和处理。数据元素可以是不可分割的“原子”型数据元素,如整数 3,字符 S 等;也可以是由多个款项构成的数据元素,其中每个款项被称为一个“数据项”。例如描述一个学生信息的数据元素可由下列 6 个数据项组成。其中的出生日期又可以由“年”、“月”和“日”三个数据项组成,因此称“出生日期”为“组合项”,而其他不可分割的数据项为“原子项”。数据项是数据结构中讨论的最小单位。

姓名 学号 性别 班号 出生日期 入学成绩

关键字是指能识别一个或多个数据元素的数据项。若能起唯一识别作用,则称之为主关键字,否则称之为次关键字。在由多个数据项构成的数据元素中必定存在关键字。

数据对象是具有相同特性的数据元素的集合,如整数、实数等。它是数据的一个子集。

1.2.2 数据结构

数据不仅存入计算机,而且必须能被正确地处理,这就要求数据元素之间的关系必须被反映出来,称这种关系为结构,因此,数据结构是一堆数据元素和这些数据元素之间关系的总和以及作用于这些元素或者结构上的一些函数或运算。

例如,可以用有序实数对表示复数, $\langle x, y \rangle$ 表示 $x + yi$ 。 x 和 y 存在前后顺序关系。

又如,可以用下述数据结构来描述 2 行 3 列的矩阵:它是一个含 6 个数据元素 $\{a_1, a_2, a_3, a_4, a_5, a_6\}$ 的集合,且集合上存在“行关系”和“列关系”两个次序关系,其中行关系为 $\{\langle a_1, a_2 \rangle, \langle a_2, a_3 \rangle, \langle a_4, a_5 \rangle, \langle a_5, a_6 \rangle\}$,列关系为 $\{\langle a_1, a_4 \rangle, \langle a_2, a_5 \rangle, \langle a_3, a_6 \rangle\}$ 。由此可见,即使数据元素集合相同,若其上的关系不同,则构成的数据结构也不同。

再如管理工作中人和人之间的关系是一种层次关系,如图 1.3 所示。

按关系或结构分,数据结构可归结为以下 4 类:集合结构、线性结构、树形结构和图状结构。如图 1.4 所示,集合结构的元素之间是松散关系;线性结构的元素之间是一一对一的关系;树形结构的元素之间是一对多的关系;图状结构的元素之间是多对多的关系。

数据结构包含如下三个方面内容。

1. 逻辑结构:表示数据元素之间的逻辑关系(是元素之间的抽象关系,与具体实现无关)。 $B = (E, R)$ 元素取自集合 E ,数据结构中的元素间有关系 R 。

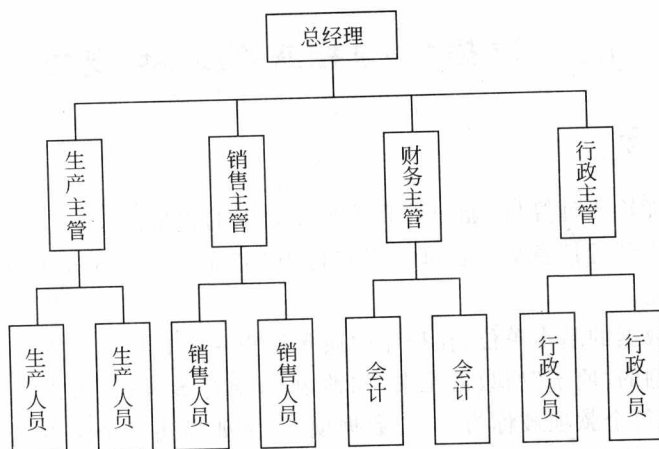


图 1.3 公司人员关系

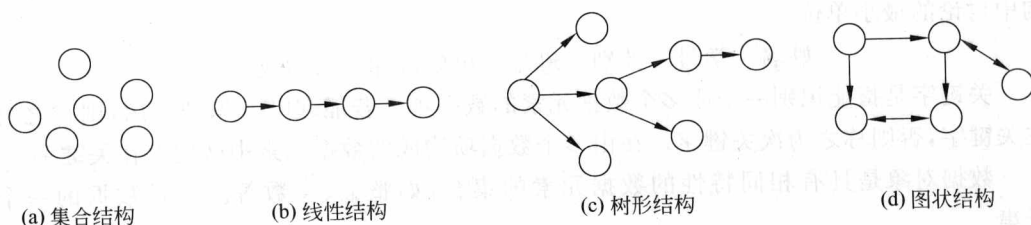


图 1.4 数据结构类型

2. 物理结构：数据的逻辑结构在计算机存储器中的映射(或表示),称为存储结构或存储表示。可采用顺序存储或链式存储两种方法,顺序存储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,通过存储地址相邻,反映逻辑上的相邻关系。因此,只要知道基地址,便可知道任何一个元素的地址;链式存储是元素可以分别存储,通过附设指针,反映元素之间的逻辑关系。

3. 作用于数据结构上的各种操作。如检索元素、插入元素和删除元素等。

作用于不同的数据结构上的操作集不同,但下列操作必不可缺。

- (1) 结构的生成。
- (2) 结构的销毁。
- (3) 在结构中查找满足规定条件的数据元素。
- (4) 在结构中插入新的数据元素。
- (5) 删除结构中已经存在的数据元素。
- (6) 遍历。

另外,在具体应用中,依据所采用的程序设计语言,可采用具体方式和技术实现某种数据结构。

本课程讨论的主要数据结构包括线性表、堆栈/队列、字符串、数组、广义表、树/二叉树、索引、哈希表和图等。

1.3 算法及其描述和分析

1.3.1 算法及其设计原则

算法是对问题求解过程的一种描述,是为解决一个或一类问题给出的一个确定的、有限长的操作序列。严格说来,一个算法必须满足以下5个重要特性。

(1) 有穷性:对于任意一组合法的输入值,在执行有穷步骤之后一定能结束。

(2) 确定性:对于每种情况下所应执行的操作,在算法中都有确切的规定,使算法的执行者或阅读者都能明确其含义及如何执行。并且在任何条件下,算法都只有一条执行路径。

(3) 可行性:算法中的所有操作都可以通过已经实现的基本操作,经过有限次运算来实现。

(4) 有输入:作为算法加工对象的量值,通常体现为算法中的一组变量。但有些算法的字面上可以没有输入,实际上已被嵌入算法之中。

(5) 有输出:它是一组与输入有确定关系的量值,是算法进行信息加工后得到的结果,这种确定关系即为算法的功能。

在进行算法设计时,通常应考虑以下原则。

(1) 设计的算法必须是“正确的”,所谓算法是正确的,除了应该满足算法说明中写明的“功能”之外,应对各组典型的带有苛刻条件的输入数据得出正确的结果。

(2) 应有很好的“可读性”,在算法是正确的前提下,算法的可读性是摆在第一位的,这在当今大型软件需要多人合作完成的环境下是至关重要的。另一方面,晦涩难读的程序易于隐藏错误而难以调试。

(3) 还必须具有“健壮性”,算法的健壮性指的是,算法应对非法输入的数据作出恰当反映或进行相应处理。一般情况下,应向调用它的函数返回一个表示错误或错误性质的值。

(4) 应考虑所设计的算法具有“高效率与低存储量”。算法的效率指的是算法的执行时间,算法的存储量指的是算法执行过程中所需最大存储空间。

1.3.2 算法的描述

本教材讨论的数据结构和算法主要是C语言,但不拘泥于某个具体的C语言,读者又容易转换成可以上机调试的C程序或C++程序。为了使算法的描述和讨论简明清晰,个别语句采用类C语言。

(1) 数据结构的表示(存储结构)都用类型定义(typedef)的方式描述。基本数据元素类型约定为DataType或ElemType,由用户在使用该数据类型时再自行具体定义。

(2) 基本操作的算法都用以下形式的函数描述。

函数类型 函数名(函数参数表)