

普通高校本科计算机专业

特色

教材精选

数据结构(C++版)

王红梅 胡明 王涛 编著

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



清华大学出版社

普通高校本科计算机专业**特色**教材精选

数据结构(C++版)

王红梅 胡明 王涛 编著

清华大学出版社
北京

内 容 简 介

数据结构是计算机专业教学计划中的核心课程,也是计算机及相关专业考研和水平等级考试的必考科目。要从事和计算机科学与技术相关的工作,尤其是计算机应用领域的开发和研制工作,必须具备坚实的数据结构基础。本书介绍了学习数据结构所用到的预备知识,叙述了数据结构、算法以及抽象数据类型概念,介绍了线性表、栈、队列和串、数组和广义表、树和二叉树、图等常用数据结构,讨论了常用的查找、排序和索引技术,给出了较多的数据结构的应用实例,最终通过一个案例将书中所有数据结构贯穿起来。

本书内容丰富,层次清晰,讲解深入浅出,可作为计算机及相关专业本、专科数据结构课程的教材,也可供从事计算机软件开发和应用的工程技术人员阅读、参考。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

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。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

数据结构(C++版)/王红梅,胡明,王涛编著. —北京:清华大学出版社,2005.7

(普通高校本科计算机专业特色教材精选)

ISBN 7-302-11258-4

I. 数… II. ①王…②胡…③王… 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(2005)第 068871 号

出 版 者: 清华大学出版社

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

社 总 机: 010-62770175

地 址: 北京清华大学学研大厦

邮 编: 100084

客户服务: 010-62776969

责任编辑: 袁勤勇

印 刷 者: 北京季蜂印刷有限公司

装 订 者: 北京市密云县京文制本装订厂

发 行 者: 新华书店总店北京发行所

开 本: 185×260 印张: 20.75 字数: 484 千字

版 次: 2005 年 7 月第 1 版 2005 年 7 月第 1 次印刷

书 号: ISBN 7-302-11258-4/TP·7427

印 数: 1~4000

定 价: 33.00 元(含光盘)

编审委员会

主任：蒋宗礼

副主任：李仲麟 何炎祥

委员：（排名不分先后）

王向东 宁 洪 朱庆生 吴功宜 吴 跃

张 虹 张 钢 张为群 余雪丽 陈志国

武 波 孟祥旭 孟小峰 胡金初 姚放吾

原福永 黄刘生 廖明宏 薛永生

秘书长：王听讲

出版说明

INTRODUCTION

在我国高等教育逐步实现大众化后,越来越多的高等学校将会面向国民经济发展的第一线,为行业、企业培养各级各类高级应用型专门人才。为此,教育部已经启动了“高等学校教学质量和教学改革工程”,强调要以信息技术为手段,深化教学改革和人才培养模式改革。如何根据社会的实际需要,根据各行各业的具体人才需求,培养具有特色显著的人才,是我们共同面临的重大问题。具体地说,培养具有一定专业特色的和特定能力强的计算机专业应用型人才则是计算机教育要解决的问题。

为了适应 21 世纪人才培养的需要,培养具有特色的计算机人才,急需一批适合各种人才培养特点的计算机专业教材。目前,一些高校在计算机专业教学和教材改革方面已经做了大量工作,许多教师在计算机专业教学和科研方面已经积累了许多宝贵经验。将他们的教研成果转化为教材的形式,向全国其他学校推广,对于深化我国高等学校的教学改革是一件十分有意义的事。

清华大学出版社在经过大量调查研究的基础上,决定组织编写一套《普通高校本科计算机专业特色教材精选》。本套教材是针对当前高等教育改革的新形势,以社会对人才的需求为导向,主要以培养应用型计算机人才为目标,立足课程改革和教材创新,广泛吸纳全国各地的高等院校计算机优秀教师参与编写,从中精选出版确实反映计算机专业教学方向的特色教材,供普通高等院校计算机专业学生使用。

本套教材具有以下特点:

1. 编写目的明确

本套教材是在深入研究各地各学校办学特色的基础上,面向普通高校的计算机专业学生编写的。学生通过本套教材,主要学习计算机科学与技术专业的基本理论和基本知识,接受利用计算机解决实际问题的基本训练,培养研究和开发计算机系统,特别是应用系统的基本能力。

2. 理论知识与实践训练相结合

根据计算学科的三个学科形态及其关系,本套教材力求突出学科的理论与实践紧密结合的特征,结合实例讲解理论,使理论来源于实践,又进一步指导实践。学生通过实践深化对理论的理解,更重要的是使学生学会理论方法的实际运用。在编写教材时突出实用性,并做到通俗易懂,易教易学,使学生不仅知其然,知其所以然,还要会其如何然。

3. 注意培养学生的动手能力

每种教材都增加了能力训练部分的内容,学生通过学习和练习,能比较熟练地应用计算机知识解决实际问题。既注重培养学生分析问题的能力,也注重培养学生解决问题的能力,以适应新经济时代对人才的需要,满足就业要求。

4. 注重教材的立体化配套

大多数教材都将陆续配套教师用课件、习题及其解答提示,学生上机实验指导等辅助教学资源,有些教材还提供能用于网上下载的文件,以方便教学。

由于各地区各学校的培养目标、教学要求和办学特色均有所不同,所以对特色教学的理解也不尽一致,我们恳切希望大家在使用教材的过程中,及时地给我们提出批评和改进意见,以便我们做好教材的修订改版工作,使其日趋完善。

我们相信经过大家的共同努力,这套教材一定能成为特色鲜明、质量上乘的优秀教材,同时,我们也希望通过本套教材的编写出版,为“高等学校教学质量和教学改革工程”作出贡献。

清华大学出版社

前言

PREFACE

数据结构是计算机及相关专业的一门重要的专业基础课，也是计算机及相关专业考研和水平等级考试的必考科目，而且正逐渐发展成为众多理工专业的热门选修课。它所讨论的知识内容和提倡的技术方法，无论对进一步学习计算机领域的其他课程，还是对从事软件工程的开发，都有着不可替代的作用。

数据结构课程知识丰富，内容抽象，学习量大，隐藏在各部分内容中的方法和技术多。本书的编写者长期从事数据结构课程的教学，对课程的教学特点和难点有比较深切的体会。在总结讲授数据结构课程的基础上，对多年来形成的数据结构课程的教学内容进行了合理的剪裁和重组，既强调数据结构的原理和方法，又注重其实践性与实用性。

本书具有以下特色：

- 系统性强 注重与前导课的知识衔接，知识单元的前驱和后继关系安排合理，每一章的内容主线清晰。
- 启发性强 精心安排的人物小传和思想火花，以及具有启发性的习题和思考题，引导学生的思维过程。
- 实用性强 随书光盘给出了本书所有算法的 C++ 程序实现，提供了学生电子笔记、算法和图解素材、英汉词汇表。
- 深入浅出 采用分析、伪代码、C++ 描述三级模式讲解算法，并设计了 250 多个插图，降低了理解问题的复杂性。
- 案例驱动 每种数据结构给出若干应用实例，提倡从实用性和实践性的角度学习数据结构，并通过一个案例将书所有的数据结构贯穿起来。

本书由三个模块构成，第一个模块是基础知识，介绍了学习数据结构的预备知识以及数据结构的基本概念；第二个模块是基本数据结构，介绍了线性表、栈、队列和串、数组和广义表、树和二叉树、图等常用数据结构；第三个模块是查找与排序技术，讨论了常用的查找技术、排序技术和索引技术。本书的随书光盘包括以下内容：每种数据结构的 C++

实现, 案例设计与实现, 电子笔记, 算法素材, 图解素材, 英汉词汇表等。

参加本书编写的还有于德海、徐立新老师, 研究生李娜、陈玥、李洋、闵聚、何文华参与了本书的案例设计并制作了光盘, 李万龙教授对本书的编写提出了很多有益的建议, 在此表示感谢。

由于作者的水平有限, 书稿虽几经修改, 仍难免有缺点和错误。热忱欢迎同行专家和读者批评指正, 使本书在使用中不断改进、日臻完善。

作 者
2005 年 3 月

目 录

CONTENTS

第 0 章 预备知识	1
0.1 数学预备知识	1
0.1.1 常用数学术语	1
0.1.2 对数	1
0.1.3 级数求和	2
0.2 常用数学证明方法	3
0.2.1 反证法	3
0.2.2 数学归纳法	3
0.3 离散数学预备知识	4
0.3.1 集合	4
0.3.2 谓词	6
0.3.3 关系	6
0.4 C++ 程序设计语言预备知识	7
0.4.1 程序结构	7
0.4.2 变量、常量与数据类型	8
0.4.3 控制语句	13
0.4.4 函数	14
0.4.5 继承与派生	19
0.4.6 多态与虚函数	20
0.4.7 模板	21
0.4.8 动态存储分配	22
0.4.9 输入与输出	23
0.4.10 异常处理	23
第 1 章 绪论	27
1.1 数据结构的兴起和发展	27
1.2 数据结构的研究对象	29

1.3	数据结构的基本概念	31
1.3.1	数据结构	31
1.3.2	数据结构的访问接口	33
1.3.3	抽象数据类型	33
1.4	算法及算法分析	35
1.4.1	算法	35
1.4.2	算法分析	38
1.5	案例综述	42
	习题 1	45
	思考题 1	47
第 2 章	线性表	49
2.1	线性表的逻辑结构	49
2.1.1	线性表的定义	49
2.1.2	线性表的抽象数据类型定义	50
2.2	线性表的顺序存储结构及实现	51
2.2.1	线性表的顺序存储结构——顺序表	51
2.2.2	顺序表的实现	52
2.3	线性表的链接存储结构及实现	57
2.3.1	线性表的链接存储结构——单链表	58
2.3.2	单链表的实现	59
2.4	顺序表和单链表的比较	66
2.4.1	时间性能比较	66
2.4.2	空间性能比较	66
2.5	线性表的其他存储方法	67
2.5.1	循环链表	67
2.5.2	双链表	68
2.5.3	静态链表	69
2.5.4	间接寻址	70
2.6	应用举例	71
2.6.1	顺序表的应用举例——符号表	71
2.6.2	单链表的应用举例——一元多项式求和	72
2.6.3	高校学籍管理	75
	习题 2	76
	思考题 2	79
第 3 章	特殊线性表——栈、队列和串	81
3.1	栈	81

3.1.1 栈的逻辑结构	81
3.1.2 栈的顺序存储结构及实现	83
3.1.3 栈的链接存储结构及实现	88
3.1.4 顺序栈和链栈的比较	89
3.2 队列	90
3.2.1 队列的逻辑结构	90
3.2.2 队列的顺序存储结构及实现	91
3.2.3 队列的链接存储结构及实现	94
3.2.4 循环队列和链队列的比较	97
3.3 串	97
3.3.1 串的逻辑结构	97
3.3.2 串的存储结构	99
3.3.3 模式匹配	100
3.4 应用举例	104
3.4.1 栈的应用举例——递归	104
3.4.2 队列的应用举例——火车车厢重排	107
3.4.3 串的应用举例——恺撒密码	109
3.4.4 高校实验任务安排问题	110
习题 3	111
思考题 3	113
第 4 章 广义线性表——多维数组和广义表	115
4.1 多维数组	115
4.1.1 数组的定义	115
4.1.2 数组的存储结构与寻址	117
4.2 矩阵的压缩存储	118
4.2.1 特殊矩阵的压缩存储	119
4.2.2 稀疏矩阵的压缩存储	121
4.3 广义表	127
4.3.1 广义表的逻辑结构	127
4.3.2 广义表的存储结构及实现	129
4.4 应用举例	132
4.4.1 数组的应用举例——魔方阵	132
4.4.2 本科生选导师问题	134
习题 4	135
思考题 4	137

第 5 章 树和二叉树	139
5.1 树的逻辑结构	139
5.1.1 树的定义和基本术语	139
5.1.2 树的抽象数据类型定义	142
5.1.3 树的遍历操作	143
5.2 树的存储结构	144
5.2.1 双亲表示法	144
5.2.2 孩子表示法	145
5.2.3 双亲孩子表示法	147
5.2.4 孩子兄弟表示法	147
5.3 二叉树的逻辑结构	148
5.3.1 二叉树的定义	148
5.3.2 二叉树的基本性质	150
5.3.3 二叉树的抽象数据类型定义	153
5.3.4 二叉树的遍历操作	154
5.4 二叉树的存储结构及实现	156
5.4.1 顺序存储结构	156
5.4.2 二叉链表	157
5.4.3 三叉链表	166
5.4.4 线索链表	167
5.5 树、森林与二叉树的转换	171
5.6 应用举例	175
5.6.1 二叉树的应用举例——哈夫曼树及哈夫曼编码	175
5.6.2 树的应用举例——8 枚硬币问题	179
5.6.3 高校学生会组织机构的管理	180
习题 5	182
思考题 5	184
第 6 章 图	185
6.1 图的逻辑结构	185
6.1.1 图的定义和基本术语	185
6.1.2 图的抽象数据类型定义	189
6.1.3 图的遍历操作	191
6.2 图的存储结构及实现	194
6.2.1 邻接矩阵	194
6.2.2 邻接表	197
6.2.3 十字链表	202
6.2.4 邻接多重表	202

6.2.5	边集数组	203
6.2.6	图的存储结构的比较	204
6.3	图的连通性	205
6.3.1	无向图的连通性	205
6.3.2	有向图的连通性	205
6.3.3	生成树和生成森林	206
6.4	应用举例	206
6.4.1	最小生成树	206
6.4.2	最短路径	211
6.4.3	AOV 网与拓扑排序	216
6.4.4	AOE 网与关键路径	220
6.4.5	校园最短路径问题	223
习题 6		224
思考题 6		227
第 7 章	查找技术	229
7.1	概 述	229
7.1.1	查找的基本概念	229
7.1.2	查找算法的性能	230
7.2	线性表的查找技术	231
7.2.1	顺序查找	231
7.2.2	折半查找	233
7.2.3	斐波那契查找	236
7.2.4	插值查找	237
7.3	树表的查找技术	238
7.3.1	二叉排序树	238
7.3.2	平衡二叉树	245
7.4	散列表的查找技术	249
7.4.1	概述	249
7.4.2	散列函数的设计	251
7.4.3	处理冲突的方法	253
7.4.4	散列查找的性能分析	257
7.4.5	开散列表与闭散列表的比较	258
习题 7		260
思考题 7		262
第 8 章	排序技术	263
8.1	概述	263

8.1.1	排序的基本概念	263
8.1.2	排序算法的性能	265
8.2	插入排序	265
8.2.1	直接插入排序	265
8.2.2	希尔排序	268
8.3	交换排序	270
8.3.1	起泡排序	270
8.3.2	快速排序	272
8.4	选择排序	277
8.4.1	简单选择排序	277
8.4.2	堆排序	279
8.5	归并排序	284
8.5.1	二路归并排序的非递归实现	284
8.5.2	二路归并排序的递归实现	288
8.6	各种排序方法的比较	289
	习题 8	292
	思考题 8	294
第 9 章	索引技术	297
9.1	索引的基本概念	297
9.2	线性索引技术	298
9.2.1	稠密索引	298
9.2.2	分块索引	299
9.2.3	多重表	300
9.2.4	倒排表	300
9.3	树形索引	301
9.3.1	2-3 树	302
9.3.2	B 树	304
9.3.3	B ⁺ 树	308
	习题 9	311
	参考文献	313

第 0 章

CHAPTER

预备知识

数据结构的先行课有高等数学、离散数学、程序设计语言等。本章介绍学习数据结构所需要的常用数学术语、对数、级数求和、常用数学证明方法等数学预备知识,集合、谓词、关系等离散数学预备知识,设计算法所用到的 C++ 程序设计语言预备知识,主要用于复习和参考。

0.1 数学预备知识

0.1.1 常用数学术语

序列:一个长度为 n 的序列是将 n 个元素按一定的线性顺序组织起来,其特点是:(1)序列中的元素有固定的顺序;(2)序列中的元素可以重复。

例如,(8, 3, 5, 2, 9, 10, 5, 9)是一个长度为 8 的序列。

向下取整:实数 x 的向下取整操作(记为 $\lfloor x \rfloor$)得到小于或等于 x 的最大整数,在 C++ 中用函数 floor 实现。例如, $\lfloor 3.4 \rfloor = 3$ 。

向上取整:实数 x 的向上取整操作(记为 $\lceil x \rceil$)得到大于或等于 x 的最小整数,在 C++ 中用函数 ceil 实现。例如, $\lceil 3.4 \rceil = 4$ 。

取模:取模是对整数进行的操作,它得到整除后非负的余数,用 $a \bmod b$ 表示,其中 b 为正整数,所得结果在 $0 \sim b-1$ 之间。在 C++ 中用运算符 % 实现^①。

例如, $8 \bmod 5 = 3$, $-8 \bmod 5 = 2$ 。

0.1.2 对数

在计算机科学中,除非有特别声明,所有对数都是以 2 为底的。在本

^① 在 C++ 中,运算符 % 的两个操作数均可以是负数。如果取模操作的任何一个操作数为负数,运算结果将随着编译器的不同而不同。例如, $-8 \% 5$ 有的编译器给出的结果是 2,而有的编译器给出的结果是 -3。如果取模操作的两个操作数均为正数,则结果是惟一的。本书中取模操作的所有操作数均为正整数。

书中,对数有两个典型的用途:

(1) 用二进制位来存储编码

许多问题都需要对一些对象进行编码,那么表示 n 个不同的对象至少需要多少二进制位呢? 答案为 $\lceil \log_2 n \rceil$ 位。例如,要存储 $\{A, B, C, D\}$ 4 个对象,编码至少需要 2 位,分别是 $\{00, 01, 10, 11\}$,要存储 1000 个不同的对象,编码至少需要 10 位。

(2) 把一个问题分解为更小的子问题,并且分解总是一分为二的。

考虑在一个有序的序列中采用折半查找,每次将待查值 k 与查找序列的中间元素进行比较,以确定下一步是在序列的前半部分还是后半部分进行查找,每比较一次都将查找序列分半,直到找到给定值,见图 0-1。一个长度为 n 的序列被逐次分半,直到序列中只有一个元素,一共需要分多少次呢? 答案为 $\lfloor \log_2 n \rfloor + 1$ 次。

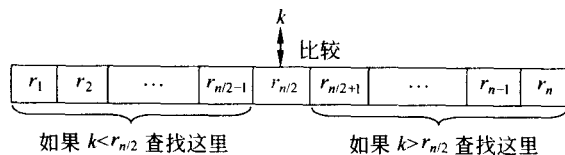


图 0-1 折半查找示意图

0.1.3 级数求和

简单地讲,级数求和就是把函数在一定范围内的值累加起来,通常表示为:

$$\sum_{i=1}^n f(i)$$

它表示对作用于某个(整数)范围内的参数 i 的函数 $f(i)$ 之值求和,函数 $f(i)$ 的参数和它的初值写在符号 $\sum$ 的下面,符号 $\sum$ 的上面表示参数的最大值。因此,这种表示法表示当 i 从 1 到 n 时对函数 $f(i)$ 的值求和,也可以表示为:

$$\sum_{i=1}^n f(i) = f(1) + f(2) + \cdots + f(n)$$

能直接计算级数和的等式称为闭合形式解。下面是本书中常用的闭合形式解:

$$\sum_{i=1}^n i = \frac{n(n+1)}{2}$$

$$\sum_{i=1}^n (n-i) = \frac{n(n-1)}{2}$$

$$\sum_{i=1}^n i^2 = \frac{2n^3 + 3n^2 + n}{6}$$

$$\sum_{i=1}^{\log n} n = n \log n$$

$$\sum_{i=1}^n 2^i = 2(2^n - 1)$$

$$\sum_{i=1}^n \frac{1}{2^i} = 1 - \frac{1}{2^n}$$

$$\sum_{i=1}^{\log n} 2^i = 2^{\log n + 1} - 2 = 2(n-1)$$

大多数程序都带有循环结构,在分析带有循环结构的程序段的时间性能时,需要把循环体的执行次数累加起来,这就是级数求和的应用。

例 0-1 分析下面程序段的执行次数。

```
for (i = 1; i <= n; i++)
    for (j = 1; j <= i; j++)
        x++;
```

解: 这个程序段的执行次数 = $\sum_{i=1}^n \sum_{j=1}^i 1 = \sum_{i=1}^n i = \frac{n(n+1)}{2}$ 。

0.2 常用数学证明方法

下面介绍本书常用的两种证明方法: 反证法和数学归纳法。

0.2.1 反证法

支持一个结论(或命题)的实例再多也不足以证明它的正确性,然而,推翻一个结论的最简单的方法就是找一个反例。反证法是一种类似于使用反例进行证明的方法。为了使用反证法证明一个结论,首先假设这个结论是错误的,然后找出由这个假设导致的逻辑上的矛盾。如果寻找矛盾的逻辑是正确的,则惟一解决矛盾的方法就是纠正我们所做结论是错误的假设,即结论是正确的。

例 0-2 证明没有最大的整数。

证明: 采用反证法。

第一步,反面假设: 假设存在一个最大的整数,记为 A 。

第二步,由假设推出矛盾: 考虑 $B = A + 1$, 因为 B 是两个整数的和,所以 B 也是整数,而 $B > A$, 因此导出矛盾。在推理过程中,惟一的漏洞就是开始时的假设是错误的,从而结论得到证明。

0.2.2 数学归纳法

设 $T(n)$ 是一个要证明的结论,其定义域为正整数。数学归纳法通过下列两个步骤来证明对于所有的 $n \geq c$ (c 是一个较小的常量), 结论 $T(n)$ 成立:

1. 初始情况: 证明 $n=c$ 时 T 成立;
2. 归纳情况: 如果在 $n-1$ 时 T 成立,则在 n 时, T 也成立。

证明初始情况通常很容易,只需要用一些较小的值代替结论中的 n , 然后应用一些简单的代数学或简单的逻辑来证明即可; 证明归纳情况有时会很难,有时会使用如下的强归纳情况。