

普通高校本科计算机专业特色教材精选 · 算法与程序设计

数据结构实践教程

主 编 徐 慧

副主编 周建美

丁卫平

陈苏蓉

章雅娟

周 洁

清华大学出版社



普通高校本科计算机专业特色教材精选·算法与程序设计

数据结构实践教程

主 编 徐 慧

副主编 周建美

丁卫平

陈苏蓉

章雅娟

周 洁

清华大学出版社
北京

内 容 简 介

本书是《数据结构》系列教材的实践教程,所有程序在 Visual Studio 6.0 下调试通过。全书分为 3 篇:第 1 篇是验证篇,给出了《数据结构》教材中所有典型数据结构的实现,是教材内容的补充;第 2 篇是设计篇,为学习者阶段学习后的实践练习和课程实验提供丰富素材;第 3 篇是综合篇,给出一些综合性的、设计量较大的题目,可以作为课程设计素材和设计指导。为了便于使用《数据结构》教材读者的学习,本书每一篇中内容安排顺序基本上与教材中的顺序一致,即线性表、栈、队列、串、数组、广义表、树、图、查找和排序等。

本书一方面可以作为《数据结构》教材的配套教材使用,起到衔接理论与实践的作用;另一方面,因其内容全面,涵盖了“数据结构”课程学习所需实践的各个方面和“数据结构”课程的主要教学内容,可作为高等院校学生学习“数据结构”课程的实践教材。本书提供了大量的源程序,对于从事计算机应用及开发的技术人员、从事数据结构和程序设计教学或学习的教师和学生具有很好的参考和指导作用。

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

数据结构实践教程 / 徐慧主编. —北京:清华大学出版社, 2010.2

(普通高校本科计算机专业特色教材精选·算法与程序设计)

ISBN 978-7-302-21476-2

I. ①数… II. ①徐… III. ①数据结构—高等学校—教材 IV. ①TP311.12

中国版本图书馆 CIP 数据核字(2010)第 009305 号

责任编辑:袁勤勇 薛 阳

责任校对:李建庄

责任印制:李红英

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

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

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

印 刷 者:北京市世界知识印刷厂

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185×260 印 张:26.5 字 数:665 千字

附光盘 1 张

版 次:2010 年 2 月第 1 版

印 次:2010 年 2 月第 1 次印刷

印 数:1~3000

定 价:35.00 元

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:035500-01

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情。

清华大学出版社在经过大量调查研究的基础上，决定组织出版一套“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，主要以培养计算机应用型人才为目标，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”做出贡献。

清华大学出版社

前言

PREFACE

“数据结构”课程内容是从长期的程序设计实践中提炼、升华而成的，主要描述现实世界实体的典型数学模型(非数值计算)及其操作在计算机中的表示和实现。它是计算机程序设计的重要理论、技术基础。因此，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计算机水平等级考试的必考科目，而且正逐渐发展为理、工、医等众多专业的热门选修课。数据结构是一门实践性很强的课程，仅通过课堂教学或自学获取理论知识是难以达到学习目的的。本书希望通过系统地、由浅入深地对数据结构学习者进行多层次实践给予切实可行的指导与帮助，使学习者学以致用。

本书分为3篇。第1篇是验证篇，给出了《数据结构》教材中所有典型数据结构的实现。通过运行这些程序，可以很好地帮助学习者理解教材中的算法，感知算法的真实，脱去算法的抽象性，顺利地从事理论过渡到实践。第2篇是设计篇，主要服务于学习者进行阶段性的实践活动或课程实验。本篇提供了符合课程实验时间较短特点的数据结构课程实验素材和指导。其中精选的实验内容不乏典型性，且每个问题强调应用教材中某一个知识点，具备一定的设计量，给出问题求解思路，让学习者动手时有着眼点，引导学习者从理论走向实践，并在实践中逐渐增强解决问题的能力。第3篇是综合篇，其中问题的解决需要应用若干个知识点，设计量较大，一般需要几个工作日完成。这些选题可以作为课程设计的内容，并提供相应指导。通过综合实验培养学习者进行复杂程序设计的能力。

如果你是学生，从第1篇可以找到教程中算法的实现。通过阅读源程序，助你从算法走向程序，为进行相近内容的程序设计提供模板和指导。通过程序运行，助你感性理解算法。实现的各种典型的数据结构，是进行其他程序设计的工具；当你想小试身手，第2篇为你提供了各种难度的设计题，挑战自己的设计和创新能力；如果小程序已经不在话下，想做一些综合应用题，第3篇为你提供了一些好的素材。

如果您是教师，第 1 篇详述的关于教程中算法的实现将是您课堂教学很好的拓展和补充，是算法的正确性证明；第 2 篇为您提供了大量课程实验素材，每个知识点均有不同难易程度的题，可适合不同层次的学生；如果课程结束后，您要指导课程设计，那第 3 篇为您提供了多层次的综合设计题，供您和学生选择。

本书由周建美老师编写了验证篇的第 8 章、设计篇的第 9、10 章、综合篇的第 2.10 节。丁卫平老师编写了设计篇的第 7、8 章、综合篇的第 2.2 节、2.4 节、2.5 节和 2.13 节。陈苏蓉、周洁老师编写了验证篇的第 9 章；章雅娟老师编写了验证篇的第 10、11 章。全书的其余部分由徐慧老师编写。管致锦老师、陈德裕老师为此书提出了许多宝贵建议。聂志浪、赵戈杰、凡刚、陈惊雷、禹杰等同学参与了许多验证程序的编写与测试工作。本书还吸取了许多前辈的成果，他们的著作给了我们许多的启示和帮助。在此，对所有为本书直接或间接地付出劳动和热情的人表示由衷地感谢！

由于作者水平有限，书中涉及的编程工作量大，难免有缺点和错误，诚请同行专家和读者批评指正，使本书在使用中不断精进。

编 者

2010 年 1 月

目 录

CONTENTS

第1篇 验证篇

第1章 绪	3
1.1 算法到程序的转换	4
1.2 验证性实践步骤	7
1.3 实验环境简介	7
1.3.1 生成项目	8
1.3.2 打开工程	10
1.3.3 生成和编辑源程序文件	10
1.3.4 在项目中加入和删除文件	11
1.3.5 调试程序	12
第2章 线性表	17
2.1 顺序存储的线性表	17
2.2 单链表	25
2.3 循环链表	33
2.4 双循环链表	41
2.5 静态链表	49
2.6 线性表的应用	58
2.6.1 集合并	58
2.6.2 多项式求和	60
第3章 栈	67
3.1 顺序栈	67
3.2 链栈	72
3.3 栈的应用	77
3.3.1 数制转换	77
3.3.2 表达式计算	79

第4章 队列	87
4.1 循环队列	87
4.2 链队	94
第5章 串	101
5.1 串类实现	101
5.2 串的模式匹配	111
第6章 数组	115
6.1 稀疏矩阵快速转置	115
6.2 矩阵加	118
6.3 矩阵乘	123
第7章 广义表	129
第8章 树与二叉树	139
8.1 二叉树	139
8.2 树	160
8.3 二叉树的线索化	172
8.3.1 先序线索化	172
8.3.2 中序线索化	179
8.3.3 后序线索化	182
8.4 最优二叉树及其应用	187
第9章 图	193
9.1 图的存储结构及图的遍历	193
9.1.1 数组表示法及图的遍历	193
9.1.2 邻接表及图的遍历	210
9.1.3 十字链表及图的遍历	225
9.1.4 邻接多重表及图的遍历	238
9.2 图的连通性问题	253
9.2.1 无向图的生成树或森林	253
9.2.2 有向图的强连通分量	258
9.2.3 最小生成树	260
9.2.4 关节点	266
9.3 有向无环图及其应用	270
9.3.1 拓扑排序	270
9.3.2 关键路径	274
9.4 最短路径	278
9.4.1 从某个源点到其余各顶点的最短路径	278
9.4.2 每一对顶点之间的最短路径	282

第 10 章 查找	287
10.1 静态查找表	287
10.2 二叉排序树	293
10.3 哈希表	300
第 11 章 内部排序	307

第 2 篇 设 计 篇

第 1 章 绪	331
1.1 实验步骤	331
1.2 实验报告格式	332
1.3 实验报告案例	333
第 2 章 线性表	335
2.1 约瑟夫问题	335
2.2 有序表合并	336
2.3 集合交运算问题	338
2.4 一元多项式求导问题	339
第 3 章 栈	341
3.1 表达式括号匹配配对判断问题	341
3.2 顺序栈的公用问题	342
3.3 迷宫问题	343
第 4 章 队列	345
4.1 轮渡模拟	345
4.2 单指针链队问题	347
4.3 显示杨辉三角形	348
4.4 栈对队列的模拟问题	349
第 5 章 串	351
5.1 统计子串在字符串中出现的次数	351
5.2 串逆置问题	352
5.3 求串中最长重复子串	352
5.4 文本文件中的单词统计	353
第 6 章 数组	355
6.1 存储格式转换问题	355
6.2 魔方问题	356
6.3 兑奖算法	357

第 7 章 树和二叉树	359
7.1 二叉树叶子结点个数计算	359
7.2 二叉树相似问题	360
7.3 二叉树任一结点的特征计算	361
7.4 压缩软件	363
第 8 章 图	367
8.1 存储结构转换问题	367
8.2 有向图的路径问题	368
8.3 邻接多重表建立无向图	369
8.4 交通指南系统	371
第 9 章 查找	373
9.1 顺序查找与折半查找的性能比较	373
9.2 开放散列与链式散列性能比较	374
9.3 动态查找——二叉排序树	375
第 10 章 排序	377
10.1 几种基本排序算法的实现.....	377
10.2 双向冒泡排序.....	378
10.3 基于单链表的直接插入排序.....	379

第 3 篇 综 合 篇

第 1 章 绪	385
1.1 综合设计内容	385
1.2 综合设计报告格式	386
第 2 章 课设习题	387
2.1 通讯录管理系统的设计与实现	387
2.2 航空客运订票系统	389
2.3 表达式求值问题	389
2.4 推箱子游戏	391
2.5 8 数码难题	392
2.6 文件目录管理与显示	393
2.7 教学计划编制	394
2.8 校园十大优秀青年评比	396
2.9 任务调度	397
2.10 修道士野人问题.....	398

2.11 食物送递服务.....	399
2.12 校园导游.....	400
2.13 中国邮路问题.....	401
附录 A 实验报告示例	403
参考文献	408

第 1 篇

验 证 篇

第1章

绪

CHAPTER

验证性实验的主要内容是教材中讲授的数据结构及应用举例涉及的算法的上机实现。学习者可以通过程序的运行,深化理解和掌握理论知识,实现算法与程序设计实现的自然衔接,为理论到实践架接桥梁。为了方便研究,本教材中给出了所有算法的源程序,其电子版可在配套光盘中找到。

本篇包含的数据结构和算法如表 1.1.1 所示。

表 1.1.1 验证性实验内容

大 类	子 类	实现的数据结构或应用
线性表	顺序表	顺序表
	链表	单链表、循环链表、双循环链表、静态链表
	应用	集合并、多项式求和
特殊线性表	栈	顺序栈、链栈
	栈应用	数制转换、表达式计算
	队列	循环队列、链队
	串	串类、串匹配
广义线性表	数组	快速转置、矩阵加、矩阵乘
	广义表	头、尾表示的广义表
树与二叉树	二叉树	二叉链表存储的二叉树
	树	孩子-兄弟存储的树
	线索二叉树	二叉树线索化、线索二叉树的遍历
	最优二叉树	创建
图	图类的实现	数组表示、邻接表表示、十字链表表示、邻接多重表表示
	图的连通性	无向图连通分量、生成树、有向图的强连通分量、最小生成树、关节点
	图的应用	拓扑排序、关键路径、Dijkstra 算法、Floyd 算法

续表

大 类	子 类	实现的数据结构或应用
查找	静态查找	顺序查找,折半查找
	动态查找	二叉排序树
	散列查找	开放定址哈希查找
排序	插入排序	直接插入排序、折半插入排序、表插入排序
	交换排序	冒泡排序、快速排序
	选择排序	简单选择排序、堆排序
	其他	归并排序、基排序

1.1 算法到程序的转换

算法是对特定问题求解步骤的一种描述。可用于描述算法的语言很多,如自然语言、框图或高级程序设计语言、类程序设计语言等。算法的最终实现用程序设计语言。算法描述语言中,最接近程序的是类程序设计语言。所谓类程序设计语言,是接近于高级语言而又不是严格的高级语言,具有高级语言的一般语句设施,但撇除了语言中某些细节的一种语言,用它来描述算法是为了更好地把注意力集中在算法处理步骤本身的描述上。一般地,用自然语言描述算法思想或算法实施步骤,用类程序设计语言描述算法,以便于最终将算法转换为程序设计语言的形式。

例:求整型数组元素中的最小值。

求解思路:采用类似“打擂”的方法。具体操作步骤如下。

step 1: 设擂主。最初把数组首元素 $A[0]$ 设为擂主,即 $A[0]$ 为最小数,以 $\min$ 表示,
 $\min = A[0]$ 。

step 2: 从次元素 $A[1]$ 直至 $A[n-1]$ 依次与 $\min$ 相比,如果比 $\min$ 小,则该数组元素为新擂主。

step 3: 输出最小数 $\min$ 。

算法的类 C++ 语言描述如下:

```
int MinNum (int A[], int n)
{
    min=A[0];
    for (i=1; i<n; i++)
        if (A[i]<min) min=A[i];
    return min;
}
```

由于类语言与程序设计语言的区别,必须认识到类语言描述的算法通常是不能直接到机器上运行的。从类语言描述的算法到算法实现,通常需要做以下几项工作。

(1) 常量、结构体、扩充数据类型的说明

算法中如果用了符号常量(如 TRUE、FALSE、OK 等),必须在程序中加以说明,如添加下列语句:

```
#define TRUE 1
#define FALSE 0
#define OK 1
```

(2) 添加库函数说明语句

程序中,如果用到 C++ 提供的库函数,需在程序中增加相应的包含语句。如:

```
#include<iostream.h>           //cout,cin
#include<process.h>             //exit()
#include<stdio.h>               //EOF,NULL
```

(3) 局部变量的添加

在类 C/C++ 中,局部变量使用前可不必说明,如上述算法中的变量 i、min 等。但在 C/C++ 中,变量必须说明后才能使用。所以,应在适当位置,添加下列变量说明语句:

```
int i, min;
```

(4) 语句的转换

类 C/C++ 中的一些语句不符合 C/C++ 语言的语法,如数据交换及一些为描述算法方便而扩充的其他语句等。

例 1:

```
a[j] ↔ a[j+1];
```

转换为:

```
x=a[j]; a[j]=a[j+1]; a[j+1]=x;
```

例 2:

```
x1=x2=2*a+b;
```

转换为:

```
x1=2*a+b; x2=x1;
```

上述求最小数的算法 MinNum 的 C++ 程序如下:

```
int MinNum (int A[],int n)
{
    int min;           //存放最小数
    min=A[0];
    for(int i=1;i<n;i++)
        if (A[i]<min) min=A[i];
    return min;
}
```