

21 世纪 高 职 高 专 规 划 教 材

计 算 机 专 业 基 础 系 列

21

数据结构实验 与实训教程

邓文华 戴大蒙 主编

清华大学出版社



21 世纪高职高专规划教材

计算机专业基础系列

数据结构实验 与实训教程

邓文华 戴大蒙 主编

清华大学出版社
北京

内 容 简 介

本书是配合计算机及相关专业“数据结构”课程而编写的。根据教学内容,并针对学生的实际情况,本教材在内容编排上共分三部分。第一部分为基础实验,给出了11个实验,包括线性结构、树型结构、图状结构、查找、排序,以及数组和字符串的操作。第二部分为综合实验,该部分实验的综合性比较强,可作为“数据结构”课程的课程设计,包括通用函数曲线作图程序、迷宫问题、全屏幕编辑程序设计、短信系统和保龄球记分系统等5个综合性实验。为了满足教学和各类学生学习与考前复习的需要,第三部分安排了10套模拟试题,并给出详细的解答。

本书内容丰富、概念清晰、实用性强,与“数据结构”课程的主要内容紧密结合,可供高职高专院校各专业学生学习、实验、课程设计和考前复习使用,也可供教师和其他专业技术人员参考。

版权所有,翻印必究。举报电话:010-62782989 13901104297 13801310933

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

数据结构实验与实训教程/邓文华,戴大蒙主编. —北京:清华大学出版社,2004.8

(21世纪高职高专规划教材·计算机应用系列)

ISBN 7-302-09020-3

I. 数… II. ①邓… ②戴… III. 数据结构—高等学校:技术学校—教材 IV. TP311.12

中国版本图书馆CIP数据核字(2004)第067879号

出 版 者:清华大学出版社 地 址:北京清华大学学研大厦

<http://www.tup.com.cn> 邮 编:100084

社 总 机:010-62770175 客户服务:010-62776969

组稿编辑:束传政

文稿编辑:田 梅

印 刷 者:北京四季青印刷厂

装 订 者:三河市李旗庄少明装订厂

发 行 者:新华书店总店北京发行所

开 本:185×230 印张:14.5 字数:296千字

版 次:2004年8月第1版 2004年8月第1次印刷

书 号:ISBN 7-302-09020-3/TP·6370

印 数:1~4000

定 价:20.00元

本书如存在文字不清、漏印以及缺页、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)62770175-3103 或(010)62795704

出版说明

高职高专教育是我国高等教育的重要组成部分,担负着为国家培养并输送生产、建设、管理、服务第一线高素质技术应用型人才的重任。

进入 21 世纪后,高职高专教育的改革和发展呈现出前所未有的发展势头,其学生规模已占我国高等教育的半壁江山,成为我国高等教育的一支重要的生力军;办学理念上,“以就业为导向”成为高等职业教育改革与发展的主旋律。近两年,教育部召开了三次产学研交流会,并启动四个专业的“国家技能型紧缺人才培养项目”,同时成立了 35 所示范性软件职业技术学院,进行两年制教学改革试点。这些举措都表明国家正在推动高职高专教育进行深层次的重大改革,向培养生产、服务第一线真正需要的应用型人才的方向发展。

为了顺应当前我国高职高专教育的发展形势,配合高职高专院校的教学改革和教材建设,进一步提高我国高职高专教育教材质量,在教育部的指导下,清华大学出版社组织出版“21 世纪高职高专规划教材”。

为推动规划教材的建设,清华大学出版社组织并成立“高职高专教育教材编审委员会”,旨在对清华版的全国性高职高专教材及教材选题进行评审,并向清华大学出版社推荐各院校办学特色鲜明的、内容质量优秀的教材选题。编审委员会的成员单位皆为教学改革成效较大、办学特色鲜明、师资实力很强的高职高专院校、普通高校以及著名企业,而教材的编写者和审定者都是从事高职高专教育第一线的优秀教师和专家。

编审委员会根据教育部最新文件政策,规划教材体系,比如部分专业的两年制教材;“以就业为导向”,以“专业技能体系”为主,突出人才培养的实践性、应用性的原则,重新组织系列课程的教材结构,整合课程体系;按照教育部制定的“高职高专教育基础课程教学基本要求”,教材的基础理论以“必要、够用”为度,突出基础理论的应用和实践技能的培养。

这套规划教材的主要特点有:

- (1) 根据岗位群设置教材系列,并成立系列教材编审委员会;
- (2) 编审委员会规划教材、评审教材;
- (3) 重点课程进行立体化建设,加强实训教材的出版,完善教学服务体系;
- (4) 教材编写者要求由有丰富教学经验和多年实践经历的教师共同组成,建立“双师型”编者体系。

本套规划教材涵盖了公共基础课、计算机、电子信息、机械、经济管理以及服务等大类

的主要课程,包括专业基础课和专业主干课。目前已经规划的教材系列名称如下:

• 公共基础课

公共基础课系列

• 计算机类

计算机基础教育系列

计算机专业基础系列

计算机应用系列

网络专业系列

软件专业系列

电子商务专业系列

• 电子信息类

电子信息基础系列

微电子技术系列

通信技术系列

电气、自动化、应用电子技术系列

• 机械类

机械基础系列

机械设计与制造专业系列

数控技术系列

模具设计与制造系列

• 经济管理类

经济管理基础系列

市场营销系列

财务会计系列

企业管理系列

物流管理系列

财政金融系列

• 服务类

旅游系列

艺术设计系列

本套规划教材的系列名称根据学科基础和岗位群方向设置,为各高职高专院校提供“自助餐”形式的教材。各院校在选择课程需要的教材时,专业课程可以根据岗位群选择系列;专业基础课程可以根据学科方向选择各类的基础课系列。例如,数控技术方向的专业课程可以在“数控技术系列”选择;数控技术专业需要的基础课程,属于计算机类课程可以在“计算机基础教育系列”和“计算机应用基础系列”选择,属于机械类课程可以在“机械基础系列”选择,属于电子信息类课程可以在“电子信息基础系列”选择。依此类推。

为更好地方便教师授课和学生学习,清华大学出版社正在建设本套教材的教学服务体系。本套教材先期选择重点课程和专业主干课程,进行立体化教材建设:加强多媒体教学课件或电子教案、素材库、学习盘、学习指导书等形式的制作和出版,开发网络课程。学校在选用教材时,可通过邮件或电话与我们联系获取相关服务,并通过与各院校的密切交流,使其日臻完善。

高职高专教育正处于新一轮改革时期,从专业设置、课程体系建设到教材编写,依然是新课题。希望各高职高专院校在教学实践中积极提出意见和建议,并向我们推荐优秀选题。反馈意见请通过 E-mail 发送到: gzgz@tup.tsinghua.edu.cn。清华大学出版社将对已出版的教材不断地修订、完善,提高教材质量,完善教材服务体系,为我国的高职高专教育出版优秀的高质量教材。

高职高专教育教材编审委员会

前言

数据结构实验与实训教程

数据结构课程是计算机专业中一门重要的专业基础课程。用计算机解决实际问题时,就要涉及数据的表示及数据的处理,而这正是数据结构课程的主要研究对象。通过这两方面内容的学习,为后续课程,特别是软件方面的课程打下了厚实的基础,同时也提供了必要的技能训练。因此,数据结构课程在计算机应用专业中具有举足轻重的作用。

要学好“数据结构”,仅仅通过课堂教学或自学获取理论知识是远远不够的,还必须加强实际动手训练。只有通过上机输入、编辑、检查、调试和运行已有的各种典型算法和已编写的程序,从成功和失败的经验中得到锻炼,才能熟练掌握和运用理论知识解决软件开发中的实际问题,达到学以致用目的。

本书与《数据结构》一书配套使用,其目的是通过实践,进一步掌握常用数据结构的基本概念及其不同的实现方法,并对在不同存储结构上实现不同的运算方法和技巧有所体会。本教材主要是为学习“数据结构”课程而编写的实训教材。本书共分三部分,第一部分为基础实验,在内容安排上给出了 11 个实验,包括了线性结构(线性结构的定义、组织形式、结构特征和类型说明,以及在两种存储方式下实现的插入、删除和按值查找的算法,循环链表、双(循环)链表的结构特点和在其上的插入、删除等操作),树型结构(二叉树的二叉链表存储方式、结点结构和类型定义,二叉树的基本运算及应用),图状结构(图的两两种存储结构的表示方法),查找(顺序查找、树表查找、散列表查找的基本思想及存储、运算的实现),排序(插入排序、冒泡排序、快速排序、直接选择排序、堆排序、归并排序和基数排序的基本思想及实现),以及数组和字符串的操作。第二部分为综合实验,这部分实验的综合性比较强,包括通用函数曲线作图程序、迷宫问题和全屏幕编辑程序设计等 5 个综合性实验,此部分可作为“数据结构”的课程设计。第三部分给出了 10 套模拟试题及其解答,目的是进一步巩固所学的理论知识。

本教材具有以下特点。

1. 每个实验题目都给出相应的 C 程序模板,在模板中填写关键语句或子程序即可上机通过,便于集中精力在算法上。
2. 实验内容多样性。习题包括基础题和提高题,以满足不同层次读者的需求。形式包括:给出程序框架要求填写关键算法,给出类似函数要求独立编写程序,给出主程序要求编写子程序,以及给出算法要求编写程序等。
3. 标准的编程风格,逐步培养良好的编程习惯。

本教材中的所有程序都在 Borland C++ 软件开发环境下调试运行通过。本书由邓文华、戴大蒙和梅志红编写。其中第一部分、第二部分及附录由邓文华和戴大蒙老师编写,第三部分由梅志红老师编写;由浙江大学计算机学院孔繁胜教授审稿。本书的程序均由戴大蒙老师上机调试通过,全书由邓文华老师最后修改定稿。

由于作者水平有限,加之编写时间仓促,错误和不当之处在所难免,敬请读者批评指正。

编 者

2004 年 5 月

目 录

数据结构实验与实训教程

第一部分 基础实验	1
实验 1 线性表的基本操作	3
实验 2 链表的基本操作	10
实验 3 栈的基本操作	18
实验 4 队列的基本操作	26
实验 5 数组的基本操作	39
实验 6 字符串的基本操作	44
实验 7 二叉树的基本操作	50
实验 8 树的遍历和 Huffman 树	57
实验 9 图的基本操作	65
实验 10 排序	73
实验 11 查找	79
基础实验答案	86
第二部分 综合实验	99
实验 1 通用函数曲线作图程序	101
实验 2 迷宫问题	106
实验 3 全屏幕编辑程序设计	120
实验 4 短信促销活动	133
实验 5 保龄球记分系统	144

第三部分 模拟试题	147
模拟试题 1	149
模拟试题 2	152
模拟试题 3	156
模拟试题 4	159
模拟试题 5	163
模拟试题 6	167
模拟试题 7	171
模拟试题 8	175
模拟试题 9	180
模拟试题 10	185
模拟试题参考答案	190
附录	217
附录 A 实验报告参考规范	217
附录 B Borland C++ 上机基本操作	218
参考文献	222

第一部分

基础实验

- 实验 1 线性表的基本操作
- 实验 2 链表的基本操作
- 实验 3 栈的基本操作
- 实验 4 队列的基本操作
- 实验 5 数组的基本操作
- 实验 6 字符串的基本操作
- 实验 7 二叉树的基本操作
- 实验 8 树的遍历和 Huffman 树
- 实验 9 图的基本操作
- 实验 10 排序
- 实验 11 查找
- 基础实验答案

实验 1

线性表的基本操作

1. 实验目的

- 1) 掌握线性表的基本运算。
- 2) 掌握顺序存储的概念,学会对顺序存储数据结构进行操作。
- 3) 加深对顺序存储数据结构的理解,逐步培养解决实际问题的编程能力。

2. 实验内容

1) 基础题

(1) 编写线性表基本操作函数。

- ① 初始化线性表 InitList(LIST * L, int ms)
- ② 向线性表指定位置插入元素 InsertList(LIST * L, int item, int rc)
- ③ 删除指定元素值的线性表记录 DeleteList1(LIST * L, int item)
- ④ 删除指定位置的线性表记录 DeleteList2(LIST * L, int rc)
- ⑤ 查找线性表中的元素 FindList(LIST * L, int item)
- ⑥ 输出线性表元素 OutputList(LIST * L)

(2) 调用上述函数实现下列操作,操作步骤如下。

- ① 初始化线性表。
- ② 调用插入函数建立一个线性表。
- ③ 在线性表中查找指定的元素。
- ④ 在线性表中删除指定值的元素。
- ⑤ 在线性表中删除指定位置的元素。
- ⑥ 遍历并输出线性表。

注意 每完成一个步骤,必须及时输出线性表元素,便于观察操作结果。

2) 提高题

(1) 要求以较高的效率实现删除线性表中元素值在 x 到 y (x 和 y 自定) 之间的所有元素。

解题思路 在线性表中设置两个初值为 0 的下标变量 i 和 j , 其中, i 为比较元素的下标, j 为赋值元素的下标。依次取线性表中下标为 i 的元素与 x 和 y 比较, 假若是 x 到 y 之外的元素, 则赋值给下标为 j 的元素。这种算法比删除一个元素后立即移动其后面的元素的效率高得多。

(2) 编程实现将两个有序的线性表进行合并, 要求同样的数据元素只出现一次。

解题思路 由于两个线性表中的元素呈有序排列, 在进行合并的时候, 依次比较, 哪个线性表的元素值小, 就先将这个元素复制到新的线性表中, 若两个元素相等, 则复制一个即可, 这样一直到其中的一个线性表结束, 然后将剩余的线性表复制到新的线性表中即可。

3. 参考程序

请在下面程序的下划线上填入适当的语句或文字。

程序 1 基础题(1), 线性表基本操作函数。

```
#include<stdio. h>
#include<stdlib. h>
#include<alloc. h>
struct LinearList                                     /* 定义线性表结构 */
{
    int * list;                                       /* 存线性表元素 */
    int size;                                         /* 存线性表长度 */
    int MAXSIZE;                                     /* 存 list 数组元素个数 */
};
typedef struct LinearList LIST;

void InitList(LIST * L, int ms)
/* 初始化线性表 */
{
    if( ( L->list = 1 ) == NULL ) {
        printf( "内存申请失败! \n" );
        exit( 1 );
    }
    2
    L->MAXSIZE = ms;
```

```
}
```

```
int InsertList( LIST *L, int item, int rc )
```

```
/* item 为记录值,rc 为插入位置 */
```

```
{
```

```
    int i;
```

```
    if( 3 )
```

```
/* 线性表已满 */
```

```
        return -1;
```

```
    if( rc < 0 )
```

```
/* 插入位置为 0 -> L->size */
```

```
        rc = 0;
```

```
    if( 4 )
```

```
        rc = L->size;
```

```
    for( i = L->size - 1; i >= rc; i-- )
```

```
/* 将线性表元素后移 */
```

```
        5
```

```
        L->list[rc] = item;
```

```
        L->size ++;
```

```
    return 0;
```

```
}
```

```
void OutputList( LIST *L )
```

```
/* 输出线性表元素 */
```

```
{
```

```
    int i;
```

```
    for( i = 0; 6 i++ )
```

```
        printf( "%d ", L->list[i] );
```

```
        printf( "\n" );
```

```
}
```

```
int FindList( LIST *L, int item )
```

```
/* 寻找线性表元素,其中返回  $\geq 0$  为元素位置,-1 为没找到 */
```

```
{
```

```
    int i;
```

```
    for( i = 0; i < L->size; i++ )
```

```
        if( 7 )
```

```
/* 找到相同的元素,返回位置 */
```

```
            return i;
```

```
    return -1;
```

```
/* 没找到 */
```

```
}
```

```
/* 删除指定元素值的线性表记录,返回 $\geq 0$  为删除成功 */
```

```

int DeleteList1( LIST *L, int item )
{
    int i, n;
    for( i = 0; i < L->size; i++ )
        if( item == L->list[i] )                /* 找到相同的元素 */
            break;
    if( i < L->size ) {
        for( n = i; n < L->size - 1; n++ )
            L->list[n] = L->list[n+1];
        L->size --;
        return i;
    }
    return -1;
}

int DeleteList2( LIST *L, int rc )
/* 删除指定位置的线性表记录 */
{
    8 /* 编写删除指定位置的线性表记录子程序 */
}

```

程序 2 基础题(2),主函数。

```

void main()
{
    LIST LL;
    int i, r;
    printf( "list addr=%p\tsize=%d\tMaxSize=%d\n", LL.list, LL.size, LL.MaxSize );
    InitList( &LL, 100 );
    printf( "list addr=%p\tsize=%d\tMaxSize=%d\n", LL.list, LL.size, LL.MaxSize );
    while( 1 )
    {
        printf( "请输入元素值,输入 0 结束插入操作:" );
        fflush( stdin );                /* 清空标准输入缓冲区 */
        scanf( "%d", &i );
        if( __ 1 __ )
            break;
        printf( "请输入插入位置:" );
        scanf( "%d", &r );
        InsertList( __ 2 __ );
    }
}

```

```
printf( "线性表为: " );  
    3  
}  
while( 1 )  
{  
    printf( "请输入查找元素值,输入 0 结束查找操作:" );  
    fflush( stdin ); /* 清空标准输入缓冲区 */  
    scanf( "%d", &i );  
    if( i == 0 )  
        break;  
    r = 4  
    if( r < 0 )  
        printf( "没找到\n" );  
    else  
        printf( "有符合条件的元素,位置为:%d\n", r+1 );  
}  
while( 1 )  
{  
    printf( "请输入删除元素值,输入 0 结束查找操作:" );  
    fflush( stdin ); /* 清空标准输入缓冲区 */  
    scanf( "%d", &i );  
    if( i == 0 )  
        break;  
    r = 5  
    if( r < 0 )  
        printf( "没找到\n" );  
    else {  
        printf( "有符合条件的元素,位置为:%d\n,线性表为:", r+1 );  
        OutputList( &LL );  
    }  
}  
while( 1 )  
{  
    printf( "请输入删除元素位置,输入 0 结束查找操作:" );  
    fflush( stdin ); /* 清空标准输入缓冲区 */  
    scanf( "%d", &r );  
    if( r == 0 )  
        break;
```

```

        i = 6;
        if( i < 0 )
            printf( "位置越界\n" );
        else {
            printf( "线性表为:" );
            OutputList( &LL );
        }
    }
}

```

程序 3 提高题。

```

#define X 10
#define Y 30
#define N 20
int A[N]={ 2, 5, 15, 30, 1, 40, 17, 50, 9, 21, 32, 8, 41, 22, 49, 31, 33, 18, 80, 5 };
#include<stdio. h>
void del( int * A, int * n, int x, int y )
{
    int i, j;
    for( i = j = 0; i < * n; i++ )
        if( A[i] > y || A[i] < x ) /* 若不在 x 到 y 之间,则保留 */
             $\frac{1}{2}$  = j;
}

```

```

void output( int * A, int n )
{
    int i;
    printf( "\n 数组有%d 个元素:\n", n );
    for( i = 0; i < n; i++ ) {
        printf( "%7d", A[i] );
        if( ( i + 1 ) % 10 == 0 )
            printf( "\n" );
    }
    printf( "\n" );
}

```

```

void main()

```