

普通高校本科计算机专业特色教材精选·算法与程序设计

Experiment Guidance and Exercises Analysis for Data Structure

数据结构实验指导与习题解析

李宗璞 薛琳 丁林花 张问银 编著



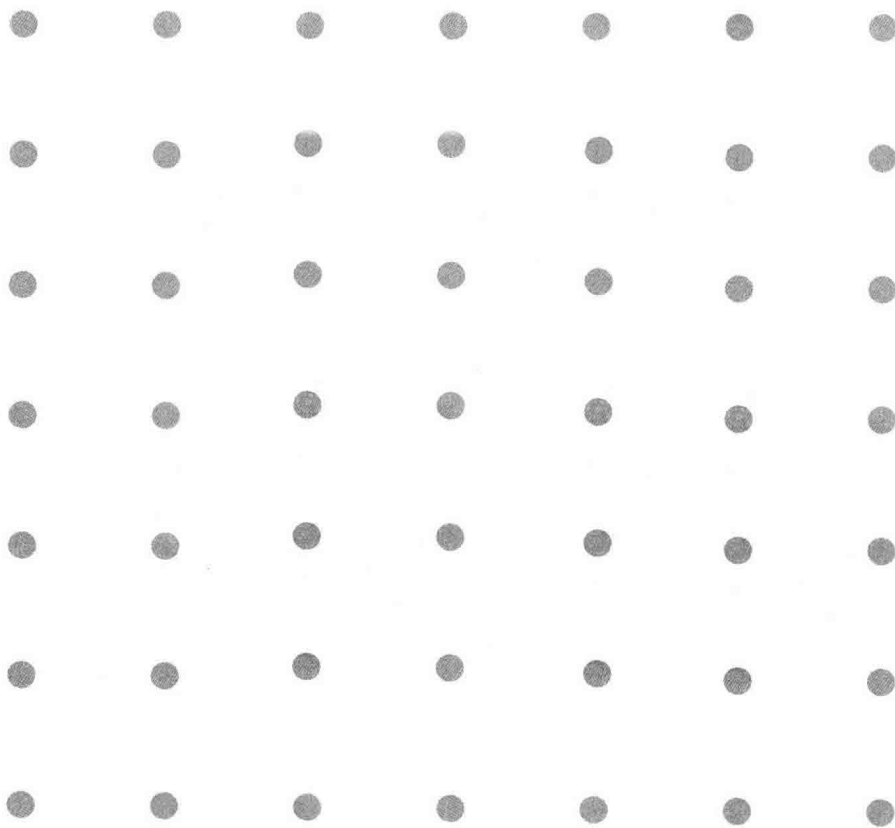
清华大学出版社

普通高校本科计算机专业特色教材精选·算法与程序设计

Experiment Guidance and Exercises Analysis for Data Structure

数据结构实验指导与习题解析

李宗璞 薛琳 丁林花 张问银 编著



清华大学出版社
北京

内 容 简 介

本书是对主教材的补充,主要包括实验部分和习题部分。

实验部分包括基础性实验和拓展性实验。基础性实验以学生应该掌握的基本知识和基本技能为依托,主要是对数据结构的基本操作进行验证性的实验,提高学生实践操作能力;拓展性实验主要是数据结构类型的应用,培养学生的创新意识和解决实际问题的分析能力。

习题部分包含知识考点、常规题型和试题解析。考点分析部分是对知识点的凝练、总结和升华,便于学生梳理知识要点;常规题型主要是对考研、升本等题型的归纳和分类,便于学生对知识的掌握,达到融会贯通、举一反三;解析部分是培养学生分析问题、解决问题以及综合应用知识的能力。

本书是《数据结构(C语言版)》的配套教材,可作为考研、升本、自学等考生的必备辅导材料。

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

数据结构实验指导与习题解析/李宗璞等编著. --北京:清华大学出版社,2016

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 978-7-302-44734-4

I. ①数… II. ①李… III. ①数据结构—高等学校—教学参考资料 IV. ①TP311.12

中国版本图书馆CIP数据核字(2016)第185986号

责任编辑:袁勤勇 李 晔

封面设计:傅瑞学

责任校对:时翠兰

责任印制:何 芊

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦A座 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

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者:北京鑫海金澳胶印有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:16.75 字 数:372千字

版 次:2016年9月第1版 印 次:2016年9月第1次印刷

印 数:1~2000

定 价:35.00元

产品编号:070224-01

前言

PREFACE

数据结构课程主要研究程序设计和软件开发过程中数据的组织、存储以及操作实现,是计算机及其相关专业的核心课程,是计算机及其相关专业的考研、升本、就业考核的课程之一。该课程理论性很强,原理和算法比较抽象,实践和应用涉及面广。对课程的学习,实验实操和习题练习是缺少不了的关键环节。

本书由实验部分和习题解析部分构成,为了更好地掌握数据结构的基本操作和广泛应用,在实验部分设计了基础性实验和拓展性实验,基础性实验部分强化学生对数据结构基本操作的掌握,拓展性实验部分培养学生对数据结构算法的实际应用能力。总之,通过实验环节主要培养学生的算法分析和设计能力,以及良好的程序设计习惯,要求学生能够掌握典型算法的设计思想及程序实现,能够根据实际问题选取合适的存储方案,设计出简洁、高效、实用的算法,为后续课程的学习及软件开发打下良好的基础。习题解析部分总结了应该掌握的基本知识和基本技能,涵盖了研究生入学考试大纲和专升本入学考试大纲中规定的必备知识点,根据考核和考试的常见题型设计了选择题、判断题、填空题、应用题和算法设计题。通过习题训练可以巩固对课程基本知识、基本技能的掌握和应用,培养学生分析问题、解决问题的能力,引导学生培养创新意识。

数据结构课程在我校开设 20 余年,师资力量雄厚,本书作者是长期从事该课程讲授和课程教学研究的骨干。多年来课程组成员勤奋努力,该课程多次获校级精品课程、校级特色课程和省级精品课程称号。以课程建设为契机,经过多年的积淀、创新和筹划,使得书稿得以成形。在撰写书稿过程中参考了大量的书籍和资料,在此对相关作者和资料提供者表示感谢。

本书的成稿受助于 2013 年度山东省高等学校精品课程项目,项目编号: 2013BK110。

作者

2016 年 4 月

目 录

CONTENTS

第1部分 实验指导

实验1 线性表	3
1.1 基础性实验	3
1.1.1 顺序表的基本操作	3
1.1.2 单链表的基本操作	8
1.2 拓展性实验	14
1.2.1 约瑟夫环	14
1.2.2 成绩管理	16
实验2 栈和队列	29
2.1 基础性实验	29
2.1.1 顺序栈的基本操作	29
2.1.2 链队列的基本操作	34
2.2 拓展性实验	39
2.2.1 表达式求值	39
2.2.2 停车场管理	41
实验3 二叉树及其操作	47
3.1 基础性实验	47
3.1.1 二叉树遍历操作	47
3.1.2 回溯法求子集	54
3.2 拓展性实验	58
3.2.1 家谱管理	58
3.2.2 图像压缩编码优化	60

实验 4 图	65
4.1 基础性实验	65
4.1.1 图的遍历(邻接矩阵存储)	65
4.1.2 图的遍历(邻接表存储)	68
4.2 拓展性实验	72
4.2.1 导航最短路径查询	72
4.2.2 公交线路查询	75
实验 5 查找和排序	81
5.1 基础性实验	81
5.1.1 查找操作	81
5.1.2 排序操作	84
5.2 拓展性实验	88
5.2.1 个人通讯录	88
5.2.2 学生管理系统	93

第 2 部分 习题解析

第 1 章 绪论	101
1.1 常规考点分析	101
1.2 常规题型	102
1.2.1 选择题	102
1.2.2 填空题	103
1.2.3 判断题	104
1.2.4 应用题	104
1.3 习题解析	104
1.3.1 选择题	104
1.3.2 填空题	106
1.3.3 判断题	107
1.3.4 应用题	108
第 2 章 线性表	111
2.1 常规考点分析	111
2.2 常规题型	113
2.2.1 选择题	113
2.2.2 填空题	117
2.2.3 判断题	118

2.2.4	算法设计题	118
2.3	习题解析	120
2.3.1	选择题	120
2.3.2	填空题	123
2.3.3	判断题	124
2.3.4	算法设计题	125
第3章	栈和队列	133
3.1	常规考点分析	133
3.2	常规题型	134
3.2.1	选择题	134
3.2.2	填空题	137
3.2.3	判断题	137
3.2.4	算法设计题	138
3.3	习题解析	139
3.3.1	选择题	139
3.3.2	填空题	142
3.3.3	判断题	143
3.3.4	算法设计题	144
第4章	串	155
4.1	常规考点分析	155
4.2	常规题型	155
4.2.1	选择题	155
4.2.2	填空题	156
4.2.3	应用题	156
4.3	习题解析	157
4.3.1	选择题	157
4.3.2	填空题	157
4.3.3	应用题	158
第5章	多维数组和广义表	159
5.1	常规考点分析	159
5.2	常规题型	159
5.2.1	选择题	159
5.2.2	填空题	161
5.3	习题解析	161
5.3.1	选择题	161

5.3.2	填空题	162
第6章	树和二叉树	165
6.1	常规考点分析	165
6.2	常规题型	166
6.2.1	选择题	166
6.2.2	填空题	173
6.2.3	判断题	174
6.2.4	应用题	175
6.2.5	算法设计题	176
6.3	习题解析	176
6.3.1	选择题	176
6.3.2	填空题	182
6.3.3	判断题	184
6.3.4	应用题	187
6.3.5	算法设计题	189
第7章	图	195
7.1	常规考点分析	195
7.2	常规题型	196
7.2.1	选择题	196
7.2.2	填空题	199
7.2.3	判断题	200
7.2.4	应用题	200
7.2.5	算法设计题	202
7.3	习题解析	203
7.3.1	选择题	203
7.3.2	填空题	205
7.3.3	判断题	206
7.3.4	应用题	206
7.3.5	算法设计题	210
第8章	查找	217
8.1	常规考点分析	217
8.2	常规题型	218
8.2.1	选择题	218
8.2.2	填空题	221
8.2.3	判断题	221

8.2.4	应用题	222
8.2.5	算法设计题	223
8.3	习题解析	224
8.3.1	选择题	224
8.3.2	填空题	227
8.3.3	判断题	227
8.3.4	应用题	228
8.3.5	算法设计题	233
第9章	排序	237
9.1	常规考点分析	237
9.2	常规题型	238
9.2.1	选择题	238
9.2.2	填空题	241
9.2.3	判断题	242
9.2.4	应用题	242
9.2.5	算法设计题	243
9.3	习题解析	243
9.3.1	选择题	243
9.3.2	填空题	247
9.3.3	判断题	248
9.3.4	应用题	249
9.3.5	算法设计题	250
	参考文献	255



第 1 部分

实 验 指 导

实验 1

线性表

EXPERIMENT

1.1 基础性实验

1.1.1 顺序表的基本操作

一、实验目的

1. 熟练掌握线性表的基本操作在顺序存储上的实现；重点巩固和体会线性表顺序存储结构上的基本操作；

2. 通过本章实验，帮助学生加深对 C 语言使用（特别是函数的调用过程、参数传递方式、结构体类型的应用等各种基本操作）的理解。

二、实验内容

1. 输入一组整型元素序列，建立顺序表，元素的值随机输入；
2. 实现该顺序表的插入；
3. 实现该顺序表的删除；
4. 实现该顺序表的查找；
5. 实现该顺序表的排序（从小到大）。

三、实验说明

1. 问题描述

输入一组整型元素序列，建立顺序表。并完成在该顺序表的插入和删除以及查找等操作。

2. 数据描述

类型定义（本书配套教材上的类型定义）：

```
#define MAXSIZE 100 //线性表可能达到的最大长度
typedef struct {
    ElemType *elem //线性表占用的数组空间
    int length; //元素的实际长度
    int maxsize; //最大长度
} SqList;
```

注意：要想想对于我们的实验题目，数据元素的类型应该是怎样的？不能用 ElemType。

3. 算法描述

在该程序中编写构造顺序表的函数、插入函数、删除函数、查找函数、排序函数，在主函数中调用这些函数，测试结果是否正确。

1) 构造函数

输入的任意 n 个数据，建立顺序表。

注意：顺序表是结构体，不能通过函数值返回，要放到参数中返回（想一想，为什么）。

2) 插入函数

在顺序表第 i 个元素之前插入新的元素 e 。

算法思路：

(1) 进行合法性检查：

i 是否合法及顺序表是否有空闲空间。

(2) 若合法，则

① 将 $a_i \sim a_n$ 顺序向下移动，为新元素让出位置；

② 将 e 置入空出的第 i 个位置；

③ 修改实际长度。

3) 删除函数

删除顺序表 L 的第 i 个元素。

算法思路：

(1) 进行合法性检查：

i 是否合法。

(2) 若合法，则

① 将 $a_{i+1} \sim a_n$ 顺序向上移动；

② 修改实际长度。

4) 查找函数

在顺序表中查询第一个满足判定条件的数据元素，若存在，则返回它的位序，否则返回 0。

算法思路：

(1) 初始化；

(2) 扫描顺序表；

(3) 判断查找成功还是失败。

5) 排序函数

将顺序表元素按照从小到大进行排序。

算法思路：

选用 C 语言中所学的选择排序或者冒泡排序，并分析所设计排序算法的时间性能（时间复杂度）。

四、注意问题

1. 插入、删除时元素的移动原因、方向及先后顺序；
2. 插入操作和删除操作中 i 的合法性检查有何不同；
3. 理解不同的函数形参与实参的传递关系。

五、算法实现示例

```
/* 顺序表操作 */
#include<stdio.h>
#include<stdlib.h>
#define MAXSIZE 100 //线性表可能达到的最大长度
typedef struct{
    int *elem; //线性表存储空间的基地址
    int length; //线性表的实际长度
}SqList; //顺序表的结构类型为 SqList
void Init_SqList(SqList *L);
void Creat_SqList(SqList *L,int n);
void output(SqList L);
int Insert_SqList(SqList *L,int i,int e);
int Delete_SqList(SqList *L,int i,int *e);
int LocateElem(SqList L,int e);
void Sort_SqList(SqList *L);
void main()
{
    SqList sl;
    int len,i,elem,e;
    Init_SqList(&sl);
    printf("请输入元素的个数: ");
    scanf("%d",&len);
    Creat_SqList(&sl,len);
    output(sl);
    printf("请输入插入位置: ");
    scanf("%d",&i);
    printf("请输入插入的元素的值: ");
    scanf("%d",&e);
    if(Insert_SqList(&sl,i,e))
        output(sl);
    else
        printf("插入位置不合法! \n");
    printf("请输入要删除的数据元素的位序: ");
    scanf("%d",&i);
    if(Delete_SqList(&sl,i,&e))
    {
        printf("被删除数据元素为%d\n",e);
    }
}
```

```
        output(s1);
    }
    else
        printf("删除位置不合法! \n");
    printf("请输入要查找的数据元素的值: ");
    scanf("%d",&elem);
    i=LocateElem(s1,elem);
    if(i) printf("所查找的数据元素在线性表中的序号为%d\n",i);
    else printf("所查找的数据元素不存在! \n");
    Sort_SeqList(&s1);
    printf("排序后的线性表元素为: ");
    output(s1);}

void Init_SeqList(SqList *L)
//顺序表初始化
{
    L->elem=(int *)malloc(sizeof(int) * MAXSIZE);    //分配初始空间
    L->length=0;
}

void Creat_SeqList(SqList *L,int n)
{//建立顺序表
    int i;
    if(n<1)                //仍然是空表
    {
        return;
    }
    if(n>100) n=100;        //只输入前 100 个
    printf("请输入%d个数据元素的值: ",n);
    for(i=0;i<n;i++)
        scanf("%d",&L->elem[i]);
    L->length=n;
}

void output(SqList L)
{
    int i;
    printf("线性表数据元素分别为: ");
    for(i=0;i<L.length;i++)
        printf("%d ",L.elem[i]);
    printf("\n");
}

int Insert_SeqList(SqList *L,int i,int e)
{
    /* 在此嵌入你的代码 */
}

int Delete_SeqList(SqList *L,int i,int *e)
{
    /* 在此嵌入你的代码 */
}
```

```
}  
int LocateElem(SqList L,int e)  
{  
    /* 在此嵌入你的代码 */  
}  
void Sort_SeqList(SqList *L)  
{  
    int i,j;  
    int lastExchangeIndex;  
    int temp;  
    i=L->length;  
    while(i>0)  
    {  
        lastExchangeIndex=0;  
        for(j=0; j<i; j++)  
            if(L->elem[j]>L->elem[j+1])  
            {  
                temp=L->elem[j+1];  
                L->elem[j+1]=L->elem[j];  
                L->elem[j]=temp;  
                lastExchangeIndex=j;    //记下进行交换的记录位置  
            }  
        i=lastExchangeIndex;  
    }  
}
```

六、测试用例

运行输入数据,如图 1-1-1 所示。

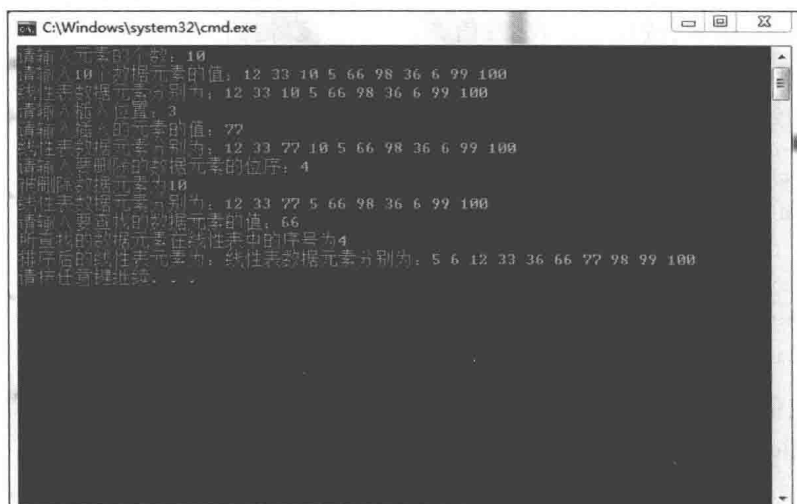


图 1-1-1 顺序表测试用例

七、思考与提高

1. 在上述示例代码中,冒泡排序尝试用标志位记录一趟排序中是否发生交换。
2. 建立顺序表,顺序表中的元素为整型并且为有序排列,要求删除表中相同的数据元素,相同结点仅保留一个,编写程序实现。

1.1.2 单链表的基本操作

一、实验目的

1. 熟练掌握线性表的基本操作在链式存储上的实现;重点巩固和体会线性表链式存储结构上的基本操作;
2. 通过本章实验帮助学生加深对指针和结构体的使用的理解。

二、实验内容

1. 建立一个带头结点的单链表,结点的值域为字符型数据,要求将用户输入的数据按头插入法来建立带头结点的单链表;
2. 实现该单链表在第 i 个元素之前插入操作;
3. 实现该单链表的删除第 i 个元素操作;
4. 实现该单链表的按序号查找操作;
5. 实现该单链表的按关键字查找操作;
6. 实现求单链表长度操作。

三、实验说明

1. 问题描述

输入一组字符元素序列,用头插法按输入序列顺序创建带头结点的单链表;完成在该单链表上的插入、删除、按序号查找、按值查找、求长度等操作。

2. 数据描述

单链表的结点结构除数据域外,还含有一个指针域。用 C 语言描述结点结构如下:

```
typedef char elemtype;
typedef struct LNode
{
    elemtype data;        //数据域
    struct LNode * next; //指针域
}LNode, LinkList;      //LinkList 为指向结构体 LNode 的指针类型
```

注意结点的建立方法及构造新结点时指针的变化。构造一个结点需用到 C 语言的标准函数 `malloc()`,如给指针变量 `p` 分配一个结点的地址:

```
p = (LinkList)malloc(sizeof(LNode));
```

该语句的功能是申请分配一个类型为 `LNode` 的结点的地址空间,并将首地址存入指针变