

## 《数据结构》实验实习指导书

### 一、实验实习目的

本实习是对学生进行基础和技能训练，使学生学会根据实际问题的要求设计算法的数据结构，培养、训练学生选用合适的数据结构和运用程序设计语言编写质量高、风格好的应用程序的能力。

### 二、实验实习任务

#### 实习一：线性表

目的：通过应用线性表的顺序存储结构和链式存储结构，熟悉两种结构的操作方式和各自的特点，重点掌握单链表的运算。

例题：有序线性表的合并问题。

#### 1. 顺序表结构

将顺序表定义为：typedef struct

```
{ elemtype List[MAXNUM];  
    int num;  
}qltype;
```

函数 MergeQL() 实现把非递减有序排列的线性表 la 和 lb 合并为一个新的非递减有序排列的线性表 lc。la 和 lb 单链表不被破坏。

主函数 main() 创建顺序表 la 和 lb。

具体参阅课堂讲解。

#### 2. 单链表结构

将单链表定义为：typedef struct SLNode

```
{ elemtype Data;  
    struct SLNode *Next;  
}slnodetype;
```

函数 CreateSL() 创建一个有 n 个结点的单链表 la;

函数 MergeSL() 实现把非递减有序排列的单链表 la 和 lb 合并为一个新的非递减有序排列的单链表 lc。la 和 lb 单链表不被破坏。

函数 DisplaySL()顺序显示单链表中的值。

主函数 main()起控制作用。

具体参阅课堂讲解。

## 实习二：堆栈与队列

目的：掌握堆栈和队列的各种操作和应用。

例题：实现堆栈的进栈和出栈操作。

### 1. 顺序结构

将顺序栈结构定义为：typedef struct

```
    { elemtype Stack[MAXNUM];  
      int Top;  
    }qstype;
```

初始化函数 InitiateQStack()

进栈操作函数 PushQStack()

出栈操作函数 PopQStack()

具体参见教材 P60-63。

### 2. 链式结构

将链式栈结构定义为：typedef struct SLNode

```
    { elemtype Data;  
      struct SLNode *Node;  
    }slnodetype;
```

进栈操作函数 PushLStack()

出栈操作函数 PopLStack()

具体参见教材 P63-66。

## 实习三：树和二叉树

目的：通过树和二叉树结构的应用，了解树和二叉树的非线性特征。

例题：建立一棵以二叉链表表示的完全二叉树，以中序方式遍历该二叉树。

二叉树中的结点定义为：typedef struct BtreeNode

```
    {elemtype Data;
```

```
struct BTreeNode *Lchild;
struct BTreeNode *Rchild;
}nodetype;
```

函数 Cre\_node()生成一个结点

函数 Create\_tree()生成二叉树

函数 InOrder()实现中序遍历二叉树

具体参阅课堂讲解。

实习四：图

目的：通过图操作算法的实现，了解图的非线性特征和应用。

例题：实现邻接矩阵存储结构下图的深度优先遍历算法。

定义图的结构为：typedef struct

```
{ SeqList Vertices ;
int edge[ MaxVertices ][ MaxVertices ] ;
int numOfEdges ;
} AdjMGraph ;
```

函数 CreatGraph()实现生成图的邻接矩阵的功能

函数 DepthFSearch()实现一次深度优先遍历的功能

主函数 main()通过调用 CreatGraph()、DepthFSearch()实现整图的遍历

具体参见教材 P202-215 相关内容

### 三、实验实习原理与方法

每个实习过程应该包括：问题分析、数据结构设计、算法设计、测试数据设计、上机调试及成果整理等。

### 四、实验实习的仪器设备

计算机、TC/VC 等语言工具

### 五、实验实习预习要求

每次实习上机前要求先根据课程内容和实习题，做好问题分析、数据结构设计、算法设计、测试数据设计工作，然后上机调试。

## 六、注意事项

在具体算法设计时，要注意函数之间的调用关系以及参数之间的数据传递问题，特别要注意指针的正确使用。

## 七、思考题

1. 已知线性表存放在数组 List[MAXNUM]的前 num( $0 \leq \text{num} \leq \text{MAXNUM}-1$ )个分量中，编写一算法删去该线性表中所有值为 x 的数据元素。

2. 计算不带表头结点的单链表的长度。

3. 计算带表头结点的单链表的长度。

4. 利用栈实现单链表的逆置。

5. 建立一棵以二叉链表表示的二叉树，以中序方式遍历该二叉树。

6. 建立一棵以二叉链表表示的二叉树，求它的结点数、叶结点数。

7. 建立一棵二叉排序树，实现二叉排序树上的查找。

8. 建立一棵二叉排序树，以中序方式遍历它。

9. 设计图的邻接矩阵结构，实现图的生成和输出。

10. 设计图的邻接表结构，实现图的生成和输出。

11. 设计图的邻接矩阵结构，实现图的深度优先和广度优先遍历操作。