**《数据结构与算法》实验**

**实验1**，**线性表**

1-1 设链表的存储结构如下：

typedef struct Node

{

Type data; //数据域；Type: 用户定义数据类型

struct Node \*next; //指针域

} Node, \*LinkList;

实现链表的基本操作。

1-2 实现顺序表的分级查找算法。基本要求包括：

(1)设计顺序表和索引表的存储结构。

(2)根据顺序表创建索引表。

(3)设计分级查找算法。

☞将2个算法的源代码(含头文件)一起打包成一个文件(用“**学号**-**姓名**”命名)，并上传到ftp服务器上的“**实验1**”文件夹下。

**实验2，树**

2-1 设二叉树的存储结构如下：

typedef struct BiTNode

{

Type data; //数据域；Type: 用户定义数据类型

struct BiTNode \*Lchild; //左指针域

struct BiTNode \*Rchild; //右指针域

} BiTNode, \*BiTree;

实现二叉树的基本操作和遍历操作。

2-2 设计算法实现：

(1) 构造哈夫曼树；

(2) 求解哈夫曼编码。

☞将2个算法的源代码文件一起打包成一个文件(用“**学号**-**姓名**”命名)，并上传到ftp服务器上的“**实验2**”文件夹下。

**实验3**，**图**

3-1 根据邻接矩阵实现图的基本操作，并设计图的深度优先搜索遍历算法和广度优先搜索遍历算法。

3-2 根据邻接表实现图的基本操作，并设计图的深度优先搜索遍历算法和广度优先搜索遍历算法。

☞将2个算法的源代码(源文件)一起打包成一个文件(用“**学号**-**姓名**”命名)，并上传到ftp服务器上的“**实验3**”文件夹下。

**实验4**，**图**

4-1 假设带权连通图G有n个顶点，用邻接矩阵A[n][n]表示存储结构，u为指定顶点的序号。试设计Prim算法，用于从顶点u出发构造连通图G的最小生成树。

4-2 采用邻接表存储结构，设计一个算法，判别无向图G中指定的两个顶点之间是否存在一条长度为k的简单路径。

注：简单路径是指顶点序列中不含有重复的顶点。

☞将2个算法的源代码文件一起打包成一个文件(用“**学号**-**姓名**”命名)，并上传到ftp服务器上的“**实验4**”文件夹下。

**实验5**，**查找**

5-1 给定有序整型数组A[n]和整数x，试设计一个在A中查找x的折半查找算法。

5-2 设二叉排序树采用二叉链表存储结构：

typedef struct BiTnode

{

KeyType key; //关键字域

ElemType \*otherinfo; //其它数据项(可以忽略)

struct BiTnode \*Lchild; //左指针域

struct BiTnode \*Rchild; //右指针域

} BiTnode, \*BiTree;

试设计二叉排序树的查找算法、插入算法和删除算法。

5-3哈希表设计。为班级30个人的姓氏(单字姓)设计一个哈希表，假设姓氏用汉语拼音表示。要求用除留取余法构造哈希函数，用线性探测再散列法处理冲突，平均查找长度的上限为2。

☞将3个算法的源代码文件一起打包成一个文件(用“**学号**-**姓名**”命名)，并上传到ftp服务器上的“**实验5**”文件夹下。

**实验6，排序**

6-1 荷兰国旗问题：设有一个仅由红、白、蓝三种颜色的条块组成的序列。试设计一个时间复杂度为O(n)的算法，使得这些条块按红、白、蓝的顺序排好，即排成荷兰国旗图案。

6-2 假设n个部门名称的基本数据存储在字符数组name[N][25]中，5≤n≤N≤20。试设计一个起泡排序算法，将n个部门名称按字典序重新排列顺序。

//附：随机产生n个部门名称

void Names(char A[][25],int n)

{

srand(time(0));

int i,j,k;

for(i=0;i<n;i++)

{

k=2\*(rand()%10+3); //部门字数

for(j=0;j<k;j++)

A[i][j]=rand()%30+176; //汉字区

A[i][j]='\0';

}

}

6-3 设计基于顺序表存储结构的树形选择排序算法。

☞将3个算法的源代码文件一起打包成一个文件(用“**学号**-**姓名**”命名)，并上传到ftp服务器上的“**实验6**”文件夹下。

**实验7，综合**

7-1 设计与实现南普陀、胡里山炮台、曾厝埯和厦门大学思明校区主要景点(如上弦场、芙蓉湖等)的旅游咨询系统，为游客提供游程最短的最优决策方案。基本要求包括：

(1)问题描述；

(2)需求分析；

(3)算法设计；

(4)系统实现；

(5)测试分析。

☞将算法的源代码文件一起打包成一个文件(用“**学号**-**姓名**”命名)，并上传到ftp服务器上的“**实验7**”文件夹下。