**以下是要掌握的概念：**

顺序表和链表的定义和同异点，插入、删除、求元素个数的操作

栈、队列指针初值的设定、访问特点及应用

串和子串的概念及存储方式、串的插入和删除，如何判断两个串相等，子串的定位（模式匹配）

数组的存储方式，数组中元素的地址计算，特殊矩阵的存储方法及地址计算

二叉树的定义及5个性质、二叉树的链式存储和顺序存储的区别

二叉树的遍历、线索二叉树的构造，指针和线索的相同点和不同点，

二叉树遍历算法的应用（求二叉树叶子节点数、内部节点数、二叉树的高度、二叉树每层节点的个数等)

哈夫曼树的构造、哈夫曼编码的求法、哈夫曼编码的特点

哈夫曼树的结点个数

无向图和有向图的邻接矩阵及其性质、图的链表表示法

图的最小生成树的求法（普利姆算法和克鲁斯卡尔算法）

最短路径的求法（迪杰斯特拉算法）

求出给定有向图的拓扑排序序列

顺序查找、折半查找的基本思想和平均查找长度，折半查找判定树的高度，

分块查找的基本思想

二叉排序树的定义

在一棵二叉排序树上进行中序遍历，输出的序列有什么特点？

在二叉排序树上查找和插入的基本思想

平衡树的定义、平衡因子的取值范围

散列法的基本思想、插入、查找、删除操作、装填因子

注意以下两个公式的用法：

h(k)=k mod p （注意p的取值）

Hi(K)=( H(k) + di) % m (i=1,2,……,m)

（di：再探测时的增量序列， m：散列表的长度）

插入排序、冒泡排序、选择排序、快速排序、堆排序、归并排序的基本思想、时间复杂度

插入排序、冒泡排序、选择排序的时间复杂度为O(n2),

快速排序、堆排序、归并排序的时间复杂度为O(n\*long2n)

**构造题：**

1. 设一台简易计算机有8条指令I1、……、I8，其使用频度分别为5%，25%，4%，7%，9%，12%，30%，8%，试为这8条指令设计前缀编码(哈夫曼编码)。
2. 设一元素序列 15，8，20，5，10，13，25，7， 画出按该排列顺序生成的二叉排序树，并为它构造中序线索二叉树; 计算在该树上查找时, 查找成功时的平均查找长度。

3.一维整型数组 T[1...8]中的元素有： 5, 12, 10, 8, 16, 7, 13, 21，这是大根堆吗？如不是，将其调整成大根堆，并把调整后的大根堆按顺序写在数组 T[]中。

4、给定哈希表 T[0...8]，哈希函数 H(k)=k % 7, 用线性探测法处理冲突。

1、 按顺序执行下列操作：

（1）插入 8、10、17

（2）删除 10

（3）查找 17

2、 在执行上述操作的过程中，发现了什么问题？把发现的问题详细地写下来。

3、 给出解决该问题的基本思想和方法。

1. 对下面的关键字集合{30,15,21,40,25,26,24,20,31}，若哈希表的装填因子为0.75，采用除留余数法和线性探测法处理冲突，

(1) 设计哈希函数；

(2) 画出哈希表；

(3) 计算查找成功时的平均查找长度；

6、设哈希函数为H(key)=key % 11,处理冲突的方法为链接法，试将下列关键字集合{15,7,22,21,19,16,25,17,5,26,11}依次插入到哈希表中(画出哈希表的示意图)，并计算查找成功时的平均查找长度。

**7. 对于下图所示的无向连通图，请用普利姆算法和克鲁斯卡尔算法求出其最小生成树，设从图中顶点1开始处理。（要求写出输出边的顺序）**

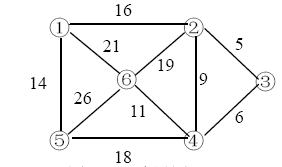
12

9 7

8 5 3

6 5 9

4



8、已知序列：12,10, 6, 25, 4，16, 28, 5，20, 8, 18，写出用直接插入排序、冒泡排序、简单选择排序、快速排序第一趟排序后的结果（假设是递增排序），并建立一个初始小根堆。

9、写出对下图进行先序遍历、中序遍历和后序遍历的结果，并说明该树是否是平衡二叉树。

A

B

C

D

E

F

G

**选择问顺序查找，折半查找的时间复杂度是什么**

10 、两个大小均为m\*n的矩阵A和B的相同位置的元素相加后得到C，即C**ij**=A**ij**+B**ij**，0≤i<n,0≤j<m。其算法如下所示。尝试分析该算法的时间复杂度是多少？

void addMat(matType C, matType A, matType B)

{ int i, j;

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

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

C[i][j]=A[i][j]+B[i][j];

}

9

1

2

3

4

5

6

7

8

11、求出右图的拓扑排序

序列。（至少3个）

9

1

2

3

4

5

6

7

8

12、依次输入表（30，15，28，20，24，10，12，68，35）中的元素构造一棵二叉排序树，并求出在等概率情况下查找成功时的平均查找长度。

13、.将下面折半查找算法补充完整。

int BinSearch(BinTable s , KeyType k)

{ /\* 在表s中用折半查找法查找关键字k，若查找成功，则函数值为该元素在表中的位置，若查找失败，返回-1。\*/

int low,mid,high;

low=0;high=s.length-1;

while(  **)**

**{**

if (s.elem[mid].key= =k) return(i); /\* 查找成功 \*/

else if(s.elem[mid].key>k)  **;**

else **;**

**}**

return(-1); /\* 查找失败 \*/

}

14、已知待散列的线性表为（6，15，14，13，19，20），散列用的一维地址空间为[0**..7**]，假定选用的散列函数是H（K）= K mod 7，若发生冲突采用线性探查法处理，试：

（1）计算出每一个元素的散列地址并在下图中填写出散列表：

` 0 1 2 3 4 5 6

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |

1. 求出在查找每一个元素概率相等情况下的平均查找长度。

**15**、以下算法采用顺序结构存储串，请将下列下划线上空白处填写完整。

typedef struct

{ char data[MAXSIZE];

int len; //串的实际长度

} SqString;

**1）将一个串t赋给串s。**

void strCopy(SqString s, SqString t)

{ int i;

for (i=0; i<t.len; i++)

{ s.data[i] = **（1）**  ; } //复制数据

s.len =  **（2）**  ;

}

**2）用于判断两个串相等。若相等则返回1，否则返回0。**

int strEqual(SqString s, SqString t)

{ int i=0;

if ( s.len <> **（3）** ) **//串长不相同时返回0**

return 0;

else

{

for (i=0; i<s.len; i++)

if ( s.data[i] <> **（4）** )

return  **（5）** ;

return 1;

}

}

**16**、.已知一棵树如图1所示，尝试回答下列问题。

1）这棵树的根结点是 ；

2）这棵树的叶子结点是 ；

3）结点E的孩子结点是 ；

4）结点E的双亲结点是 。

5）这棵树的深度是 ；

17、如图2所示的一棵二叉树，写出先序遍历序列、中序遍历序列、后序遍历序列和层次遍历序列。

18、如图7.21所示为一个有向图，求出

(1)每个顶点的入度和出度

V1

V5

V2

V4

V6

V3

(2)邻接距阵

(3)邻接表

19、给定序列：10 6 -21 5 -17 16 17 25 -4 -8 0 ，把负数放在左边，把正数放在左边。

**20、填表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **排序名称** | 直接插入排序 | 冒泡排序 | 简单选择排序 | 快速排序 | 堆排序 | 二路归并排序 |
| **时间复杂度** |  |  |  |  |  |  |
| **稳定性** |  |  |  |  |  |  |

1. 分别写出图3从V0出发按深度优先搜索遍历的得到的顶点序列和按广度优先搜索遍历得到的顶点序列。

图3 一幅图结构

1. 在数组T[1...12]中存有序列：15，10，18，9，17，4，8，3，12，6，13，把该序列调整成一个小根堆，并把该小根堆按顺序写回数组T[1...12]中。

**算法题**

1.求出任一棵二叉树的叶子结点数和内部结点数。

2.设一棵二叉树结点的信息字段是整型，判断该二叉树是否是二叉排序树。