1.     栈和队列的共同特点是( )。

A.只允许在端点处插入和删除元素

B.都是先进后出

C.都是先进先出

D.没有共同点

2.     用链接方式存储的队列，在进行插入运算时( ).

A. 仅修改头指针 　 B. 头、尾指针都要修改

C. 仅修改尾指针 D.头、尾指针可能都要修改

3.     以下数据结构中哪一个是非线性结构？( )

A. 队列 　　 B. 栈 C. 线性表 　　 D. 二叉树

4.     设有一个二维数组*A*[*m*][*n*]，假设*A*[0][0]存放位置在644(10)，*A*[2][2]存放位置在676(10)，每个元素占一个空间，问*A*[3][3](10)存放在什么位置？脚注(10)表示用10进制表示。

A．688 B．678 C．692 D．696

5.     树最适合用来表示( )。

A.有序数据元素 B.无序数据元素

C.元素之间具有分支层次关系的数据 D.元素之间无联系的数据

6.     二叉树的第k层的结点数最多为( ).

A．2k-1 B.2K+1 C.2K-1 　　　D. 2k-1

7.     若有18个元素的有序表存放在一维数组A[19]中，第一个元素放A[1]中，现进行二分查找，则查找A［3］的比较序列的下标依次为( )

A. 1，2，3 B. 9，5，2，3

C. 9，5，3 D. 9，4，2，3

8.     对n个记录的文件进行快速排序，所需要的辅助存储空间大致为

A. O（1） 　　B. O（n）　　 C. O（1og2n） D. O（n2）

9.     对于线性表（7，34，55，25，64，46，20，10）进行散列存储时，若选用H（K）=K %9作为散列函数，则散列地址为1的元素有（ ）个，

A．1 B．2 C．3 D．4

10. 设有6个结点的无向图，该图至少应有( )条边才能确保是一个连通图。

A.5 B.6 C.7 D.8

11、在一个长度为n的顺序线性表中顺序查找值为x的元素时，查找成功时的平均查找长度（即x与元素的平均比较次数，假定查找每个元素的概率都相等）为 ( )。

A n B n/2 C (n+1)/2 D (n-1)/2

12、在一个单链表中,若q所指结点是p所指结点的前驱结点,若在q与p之间插入一个s所指的结点,则执行( )。

A s→link=p→link; p→link=s; B p→link=s; s→link=q;

C p→link=s→link; s→link=p; D q →link=s; s→link =p;

13、      栈的插入和删除操作在（ ）进行。

A 栈顶 B 栈底 C 任意位置 D 指定位置

14、      由权值分别为11，8，6，2，5的叶子结点生成一棵哈夫曼树，它的带权路径长度为（ ）

A 24 B 71 C 48 D 53

15、设某数据结构的二元组形式表示为A=(D，R)，D={01，02，03，04，05，06，07，08，09}，R={r}，r={<01，02>，<01，03>，<01，04>，<02，05>，<02，06>，<03，07>，<03，08>，<03，09>}，则数据结构A是（ ）。

(A) 线性结构 (B) 树型结构 (C) 物理结构 (D) 图型结构

16、下面程序的时间复杂为（ ）

for（i=1，s=0； i<=n； i++） {t=1；for(j=1；j<=i；j++) t=t\*j；s=s+t；}

(A) O(n) (B) O(n2) (C) O(n3) (D) O(n4)

17、设指针变量p指向单链表中结点A，若删除单链表中结点A，则需要修改指针的操作序列为（ ）。

(A) q=p->next；p->data=q->data；p->next=q->next；free(q)；

(B) q=p->next；q->data=p->data；p->next=q->next；free(q)；

(C) q=p->next；p->next=q->next；free(q)；

(D) q=p->next；p->data=q->data；free(q)；

18、设有n个待排序的记录关键字，则在堆排序中需要（ ）个辅助记录单元。

(A) 1 (B) n (C) nlog2n (D) n2

19、设一组初始关键字记录关键字为(20，15，14，18，21，36，40，10)，则以20为基准记录的一趟快速排序结束后的结果为( )。

(A) 10，15，14，18，20，36，40，21

(B) 10，15，14，18，20，40，36，21

(C) 10，15，14，20，18，40，36，2l

(D) 15，10，14，18，20，36，40，21

20、设二叉排序树中有n个结点，则在二叉排序树的平均平均查找长度为（ ）。

(A) O(1) (B) O(log2n) (C) (D) O(n2)

21设无向图G中有n个顶点e条边，则其对应的邻接表中的表头结点和表结点的个数分别为（ ）。

(A) n，e (B) e，n (C) 2n，e (D) n，2e

22 设某强连通图中有n个顶点，则该强连通图中至少有（ ）条边。

(A) n(n-1) (B) n+1 (C) n (D) n(n+1)

23设有5000个待排序的记录关键字，如果需要用最快的方法选出其中最小的10个记录关键字，则用下列（ ）方法可以达到此目的。

(A) 快速排序 (B) 堆排序 (C) 归并排序 (D) 插入排序

24、下列四种排序中（ ）的空间复杂度最大。

(A) 插入排序 (B) 冒泡排序 (C) 堆排序 (D) 归并排序

**二、                   运算题（每题 6 分，共24分）**

1.        数据结构是指数据及其相互之间的\_\_\_\_\_\_\_\_\_\_\_\_\_\_。当结点之间存在M对N（M：N）的联系时，称这种结构为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。

2.        队列的插入操作是在队列的\_\_\_尾\_\_\_\_\_\_进行，删除操作是在队列的\_\_\_\_首\_\_\_\_\_\_进行。

3.        当用长度为N的数组顺序存储一个栈时，假定用top==N表示栈空，则表示栈满的条件是\_\_\_ \_\_\_(要超出才为满)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。

4.        对于一个长度为n的单链存储的线性表，在表头插入元素的时间复杂度为\_\_\_\_\_\_\_\_\_，在表尾插入元素的时间复杂度为\_\_\_\_\_\_\_\_\_\_\_\_。

5.        设W为一个二维数组，其每个数据元素占用4个字节，行下标i从0到7 ，列下标j从0到3 ，则二维数组W的数据元素共占用＿\_\_\_\_\_＿个字节。W中第6 行的元素和第4 列的元素共占用＿\_\_\_\_\_\_\_＿个字节。若按行顺序存放二维数组W，其起始地址为100，则二维数组元素W[6，3]的起始地址为＿\_\_\_\_\_\_\_\_＿。

6.        广义表A= (a,(a,b),((a,b),c)),则它的深度为\_\_\_\_\_\_\_\_\_\_\_\_，它的长度为\_\_\_\_\_\_\_\_\_\_\_\_。

7.        二叉树是指度为2的\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_树。一棵结点数为N的二叉树，其所有结点的度的总和是\_\_\_\_\_\_\_\_\_\_\_\_\_。

8.        对一棵二叉搜索树进行中序遍历时，得到的结点序列是一个\_\_\_\_\_\_\_\_\_\_\_\_\_\_。对一棵由算术表达式组成的二叉语法树进行后序遍历得到的结点序列是该算术表达式的\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。

9.        对于一棵具有n个结点的二叉树，用二叉链表存储时，其指针总数为\_\_\_\_\_\_\_\_\_\_\_\_\_个，其中\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_个用于指向孩子，\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_个指针是空闲的。

10.    若对一棵完全二叉树从0开始进行结点的编号，并按此编号把它顺序存储到一维数组A中，即编号为0的结点存储到A[0]中。其余类推，则A[ i ]元素的左孩子元素为\_\_\_\_\_\_\_\_,右孩子元素为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_，双亲元素为\_\_\_\_\_\_\_\_\_\_\_\_。

11.    在线性表的散列存储中，处理冲突的常用方法有\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_和\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_两种。

**12.** 当待排序的记录数较大，排序码较随机且对稳定性不作要求时，宜采用\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_排序；当待排序的记录数较大，存储空间允许且要求排序是稳定时，宜采用\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_排序。

**三、                   运算题（每题6分，共24分）**

1.        已知一个6×5稀疏矩阵如下所示，

|  |
| --- |
|  |

试：

（1）        写出它的三元组线性表；

（2）        给出三元组线性表的顺序存储表示。

2.        设有一个输入数据的序列是 **{** 46, 25, 78, 62, 12, 80 **}**, 试画出从空树起，逐个输入各个数据而生成的二叉排序树。

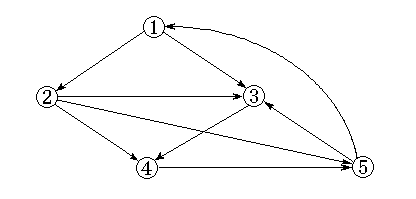
3.        对于图6所示的有向图若存储它采用邻接表，并且每个顶点邻接表中的边结点都是按照终点序号从小到大的次序链接的，试写出：

(1) 从顶点①出发进行深度优先搜索所得到的深度优先生成树；

(2) 从顶点②出发进行广度优先搜索所得到的广度优先生成树；

4.        已知一个图的顶点集V和边集E分别为：

V={1,2,3,4,5,6,7};



|  |
| --- |
| 图6 |

E={<2,1>,<3,2>,<3,6>,<4,3>,<4,5>,<4,6>,<5,1>,<5,7>,<6,1>,<6,2>,<6,5>};

若存储它采用邻接表，并且每个顶点邻接表中的边结点都是按照终点序号从小到大的次序链接的，按主教材中介绍的拓朴排序算法进行排序，试给出得到的拓朴排序的序列。

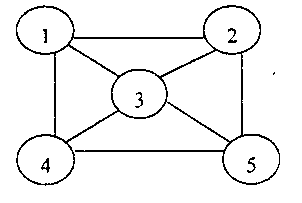
5.        在如下数组A中链接存储了一个线性表，表头指针为A [0].next，试写出该线性表。

A 0 1 2 3 4 5 6 7

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| data |  | 60 | 50 | 78 | 90 | 34 |  | 40 |
| next | 3 | 5 | 7 | 2 | 0 | 4 |  | 1 |

6. 请画出图10的邻接矩阵和邻接表。

7.        已知一个图的顶点集V和边集E分别为：



|  |
| --- |
| 图10 |

V={1,2,3,4,5,6,7}; E={(1,2)3,(1,3)5,(1,4)8,(2,5)10,(2,3)6,(3,4)15,

(3,5)12,(3,6)9,(4,6)4,(4,7)20,(5,6)18,(6,7)25};

用克鲁斯卡尔算法得到最小生成树，试写出在最小生成树中依次得到的各条边。

8.        画出向小根堆中加入数据4, 2, 5, 8, 3时，每加入一个数据后堆的变化。

9、已知一个散列表如下图所示：

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | 35 |  | 20 |  |  | 33 |  | 48 |  |  | 59 |

0 1 2 3 4 5 6 7 8 9 10 11 12

其散列函数为h(key)=key%13, 处理冲突的方法为双重散列法，探查序列为：

hi=(h(key)+\*h1(key))%m =0,1,…，m－1

其中

h1(key)=key%11+1

回答下列问题：

（1）对表中关键字35，20，33和48进行查找时，所需进行的比较次数各为多少？

（2）该散列表在等概率查找时查找成功的平均查找长度为多少？

10、有7带权结点,其权值分别为2,6,10,14,3,7,8。

(1)试以它们为叶子结点构造一棵哈夫曼树（要求按每个结点的左子树根结点的权值小于或等于右子树根结点的权值的次序构造）（3分）

2）计算该树的结点总数（3分）

3）计算其带权路径长度WPL（4分）

**四、                   阅读算法（每题7分，共14分）**

1. 1.                     int Prime(int n)

{

int i=1;

int x=(int) sqrt(n);

while (++i<=x)

if (n%i==0) break;

if (i>x) return 1;

else return 0;

}

(1)     指出该算法的功能；

(2)     该算法的时间复杂度是多少？

2.                     写出下述算法的功能：

void AJ(adjlist GL, int i, int n)

{

Queue Q;

InitQueue(Q);

cout<<i<<' ';

visited[i]=true;

QInsert(Q,i);

while(!QueueEmpty(Q)) {

int k=QDelete(Q);

edgenode\* p=GL[k];

while(p!=NULL)

{

int j=p->adjvex;

if(!visited[j])

{

cout<<j<<' ';

visited[j]=true;

QInsert(Q,j);

}

p=p->next;

}

}

}

3.     LinkList mynote(LinkList L)

{//L是不带头结点的单链表的头指针

if(L&&L->next){

q=L；L=L－>next；p=L；

S1： while(p－>next) p=p－>next；

S2： p－>next=q；q－>next=NULL；

}

return L；

}

请回答下列问题：

（1）说明语句S1的功能；

（2）说明语句组S2的功能；

（3）设链表表示的线性表为（a1,a2, …,an）,写出算法执行后的返回值所表示的线性表。

4.     void ABC(BTNode \* BT)

{

if BT {

ABC (BT->left);

ABC (BT->right);

cout<<BT->data<<' ';

}

}

该算法的功能是：

**五、                   算法填空（共8分）**

1、如下为二分查找的非递归算法，试将其填写完整。

Int Binsch(ElemType A[ ],int n,KeyType K)

{

int low=0;

int high=n-1;

while (low<=high)

{

int mid=\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_；

if (K==A[mid].key) return mid; //查找成功，返回元素的下标

else if (K<[mid].key)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_; //在左子表上继续查找

else \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_; //在右子表上继续查找

}

return -1; //查找失败，返回-1

}

 2、二叉搜索树的查找——递归算法:

bool Find(BTreeNode\* BST,ElemType& item)

{

if (BST==NULL)

return false; //查找失败

else {

if (item==BST->data){

item=BST->data;//查找成功

return ­­­­­­­­­\_\_\_\_\_\_\_\_\_\_\_;}

else if(item<BST->data)

return Find(\_\_\_\_\_\_\_\_\_\_\_\_\_\_,item);

else return Find(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_,item);

}//if

}

**六、                   编写算法（共8分）**

1、HL是单链表的头指针，试写出删除头结点的算法。

ElemType DeleFront(LNode \* & HL)

2、统计出单链表HL中结点的值等于给定值X的结点数。

int CountX(LNode\* HL,ElemType x)

3.         设计判断两个二叉树是否相同的算法。

4.         设计两个有序单链表的合并排序算法。

1. 设计在链式存储结构上交换二叉树中所有结点左右子树的算法。

6、   在链式存储结构上建立一棵二叉排序树。