**一、单项选择题**

1．计算机算法指的是( C )。

A．计算方法 B．排序方法

C．解决问题的步骤序列 D．调度方法

2．从逻辑上可以把数据结构分为( C )两大类。

A．动态结构、静态结构 B．顺序结构、链式结构

C．线性结构、非线性结构 D．初等结构、构造型结构

3．算法分析的两个方面是( B )。

A．正确性和简明性 B．空间复杂度和时间复杂度

C．可读性和文档性 D．数据复杂性和程序复杂性

4．若某线性表最常用的操作是存取任意指定序号的元素和在最后进行插入和删除运算，则利用( A )存储方式最节省时间。

A．顺序表 B．双链表

C．带头结点的双循环链表 D．单循环链表

5、双向链表中有两个指针域（llink和rlink），分别指向前驱和后继，设p指向链表中的一个结点，q指向一待插入结点，现要求在p前插入q，则正确的插入为( D )。

A．p->llink=q; q->rlink=p; p->llink->rlink=q; q->llink=p->llink;

B．q->llink=p->llink; p->llink->rlink=q; q->rlink=p; p->llink=q->rlink;

C．q->rlink=p; p->rlink=q; p->llink->rlink=q; q->rlink:=p;

D．p->llink->rlink=q; q->rlink=p; q->llink=p->llink; p->llink=q;

6．对于栈操作数据的原则是( B )。

A．先进先出 B．后进先出

C．后进后出 D．不分顺序

7．有六个元素 6，5，4，3，2，1 的顺序进栈，（ C ）不是合法的出栈序列。

A．5，4，3，6，1，2 B．4，5，3，1，2，6

C．3，4，6，5，2，1 D．2，3，4，1，5，6

8．假设以数组A[m]存放循环队列的元素,其头尾指针分别为front和rear，则当前队列中的元素个数为( A )。

A．(rear-front+m)%m B．rear-front+1

C．rear-front-1 D．(rear-front+1)%m

9．递归过程或函数调用时，处理参数及返回地址，要用一种称为( C )的数据结构。

A．队列 B．多维数组

C．栈 D．线性表

10．在一棵非空二叉树的中序遍历序列中，根结点的右边( A )。

A．只有右子树上的所有结点 B．只有右子树上的部分结点

C．只有左子树上的所有结点 D．只有左子树上的部分结点

11．最大容量为n的循环队列，队尾指针是rear，队头是front，则队空的条件是( B )。

A．(rear+1) MOD n==front B．rear==front

C．rear+1==front D．(rear-l) MOD n==front

12．一个有n个顶点的无向图最多有( D )条边。

A．n B．n(n-1)

C．2n D．n(n-1)/2

13．深度为5的二叉树至多有( A )个结点。

A．31 B．32

C．16 D．10

14．若一棵二叉树具有10个度为2的结点，5个度为1的结点，则度为0的结点个数是( B )。

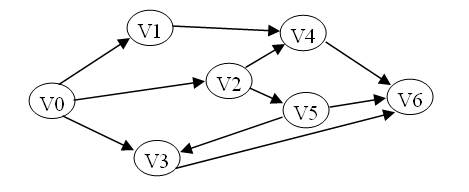
A．9 B．11

C．15 D．不确定

15．对下图所示的有向图，从顶点V0出发的深度优先遍历序列是( B )。

A．V0 V1 V2 V4 V6 V3 V5 B．V0 V1 V4 V6 V2 V5 V3

C．V0 V2 V5 V6 V1 V3 V4 D．V0 V1 V2 V3 V4 V5 V6



16．设给定权值总数有n个，其哈夫曼树的结点总数为( D )。

A．不确定 B．2n

C．2n+1 D．2n-1

17．对线性表进行折半查找时，必须要求线性表是( C )。

A．以顺序方式存储

B．以链式方式存储

C．以顺序方式存储，且结点按关键字有序排列

D．以链式方式存储，且结点按关键字有序排列

18．二叉排序树是( D )。

A．每一分支结点的度均为2的二叉树

B．中序遍历得到一升序序列的二叉树

C．按从左到右顺序编号的二叉树

D．每一分支结点的值均小于左子树上所有结点的值，大于右子树上所有结点的值

19．下列说法不正确的是( C )。

A．图的遍历是从给定的源点出发每一个顶点仅被访问一次

B．遍历的基本算法有两种：深度优先遍历和广度优先遍历

C．图的深度优先遍历不适用于有向图

D．图的深度优先遍历是一个递归过程

20．在一个包含n个顶点e条边的无向图的邻接矩阵中，零元素的个数为( D )。

A．e B．2e

C．n2-e D．n2-2e

**二、判断对错**

【 √ 】1．哈夫曼树的所有子树也都是哈夫曼树。

【 √ 】2．图中某顶点的度是所有与该顶点关联的边的数目。

【 X 】3．线性顺序存储结构不要求逻辑上相邻的数据元素在存储位置上也相邻。

【 √ 】4．假定有4个元素按照A，B，C，D的顺序入栈，入栈过程中允许出栈,则CBAD是可能的出栈序列。

【 X 】5．要想通过遍历序列还原为惟一二叉树，只需知道其中序序列即可。

【 √ 】6．栈和队列的共同点是只允许在端点处插入和删除元素。

【 √ 】7．二叉树中不存在度大于2的结点。

【 √ 】8．使用循环队列可以解决队列顺序存储时的假溢出问题。

【 X 】9．具有n个顶点的连通图至少有n条边。

【 X 】10．当图的边较少而结点较多时，求其最小生成树用Prim算法比用Kruskal算法效率更高。

**三、 填空题**

1．抽象数据类型的定义仅取决于它的一组\_\_【1】\_，而与【2】无关，即不论其内部结构如何变化，只要它的【3】不变，都不影响其外部使用。

【1】逻辑特性 【2】在计算机内部如何表示和实现无关 【3】数学特性

2．在用于表示有向图的邻接矩阵中，对第i行的元素进行累加，可得到第i个顶点的【4】度，而对第j列的元素进行累加，可得到第j个顶点的【5】度。

【4】出度 【5】入度

3．在有序表（12，24，36，48，60，72，84）中二分查找关键字72时所需进行的关键字比较次数为【6】 。

【6】2

4．将含100个结点的完全二叉树从根这一层开始，每层上从左到右依次对结点编号，根结点的编号为1。编号为49的结点X的双亲编号为 【7】 、左孩子节点编号为【8】 、右孩子节点编号为 【9】 。

【7】24 【8】98 【9】99

5．3个结点可构成 【10】 个不同形态的二叉树。

【10】5

6．如果频繁地对线性表进行插入和删除操作，该线性表应采用 【11】 存储结构。

【11】链式存储结构

7．以{ 2 ,3 ,6 ,7 ,9 ,12 }作为叶子结点的权值构造哈夫曼树，则其最短带权路径长度为 【12】 。

【12】94

8．算法的5个重要特征是 【13】 、 【14】 、【15】 、输入和输出。

【13】-【15】：有穷性 确定性 可行性

**四、按照指定功能，完成下列算法**

1、 在二叉排序树中查找关键字等于key 的元素

Status SearchBST ( BiTree T, KeyType key, BiTree f, BiTree &p)

//在根指针T所指二叉排序树中递归地查找其关键字等于key的数据元素，若查找

//成功， 则指针p指向该数据元素结点，并返回TRUE，否则指针指向查找路径上访

//问的最后一个结点并返回FALSE，指针f指向T的双亲，其初始调用值为NULL

{ if ( !T ) { 【1】 }

else if ( EQ(key,T->data.key) ) {【2】}

else if ( LT(key, T->data.key) ) 【3】

else 【4】

}

【1】return FALSE;

【2】P=T; return TRUE;

【3】return SearchBST(T->lchild, key, T, p);

【4】return SearchBST(T->rchild, key, T, p);

2、循环队列入队操作算法：

Status EnQueue\_Sq ( SqQueue &Q, ElemType e )  
{ //将元素e插入队尾  
 if ( 【5】 ) return ERROR ; //队满  
 Q.base[Q.rear] = e ; //将元素e插入队尾

【6】 //修改队尾指针

return OK;  
} //EnQueue\_Sq

【5】(Q.rear+1)%MAXSIZE= =Q.front )

【6】Q.rear = (Q.rear+1)%MAXSIZE;

3、一趟快速排序算法

int Partition (RedType& R[], int low, int high)

{ R[0] = R[low]; pivotkey = R[low].key; // 枢轴

while (low<high)

{ while (low<high && 【7】 ) 【8】 ；

R[low]=R[high];

while (low<high && 【9】 ) 【10】 ；

R[high]=R[low];

}

R[low] = R[0]; return low; // 返回枢轴所在位置

}

【7】R[high].key>=pivotkey

【8】--high;

【9】R[low].key<=pivotkey

【10】++low;

**五、简答题**

1．有一份报文共使用5个字符：a、b、c、d、e，它们出现频率依次是4、7、5、2、9，请完成如下工作：

（1）以字符出现频率为权，构造哈夫曼树（规定权值较小的结点为左子树）；

（2）求出每个字符的哈夫曼编码；

2．对于如下所示的加权无向图，画出用普里姆（Prim）算法构造最小生成树的过程中，从E顶点开始，当生成树有1条、2条、4条、6条边时的生成树。

**3**

**4**

**2**

**9**

**1**

**4**

**8**

**7**

**10**

**5**

**11**

**6**

**12**

第2题图

最小生成树的构造过程如下图所示：

**1**

**2**

**1**

**2**

**4**

**3**

**1**

**7**

**2**

**6**

**4**

**3**

**1**

3．设结点关键码的输入序列为{53，68，55，17，82，10，45}，试利用二叉排序树的插入算法，分别画出插入过程中二叉排序树有3个、5个和全部节点时的二叉排序树。

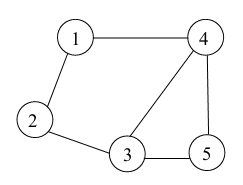
4．一棵二叉树的中序序列和后序序列分别是 DCBAEFG 和 DCBGFEA，试画出这棵二叉树，并给出该二叉树的先序遍历序列。

5．已知一个有向图如下图所示，请给出该图的：

（1）邻接矩阵；

（2）邻接表；

（3）自顶点4出发进行深度优先遍历和广度优先遍历各一个遍历顶点序列。



第3题图

六、**给出下列程序运行结果**

#include < stdio.h >

int c = 0 ;

void move( char x, int n, char z )

{

printf( "第%i步: 将%i号盘从%c移到%c\n", ++c, n, x, z ) ;

}

void hanoi( int n, char x, char y, char z )

{ if( n==1 )

move( x, 1, z ) ;

else

{

hanoi( n-1, x, z, y ) ;

move( x, n, z ) ;

hanoi( n-1, y, x, z ) ;

}

}

void main( )

{

int n = 3 ;

hanoi( n, 'a', 'b', 'c' ) ;

}

**七、算法设计**

假设以两个元素依值递增有序排列的线性表A和B分别表示两个集合（同一表中的元素值各不相同）。现要求另开辟空间构成一个线性表C，其元素为A和B中元素的交集，且表C中的元素也依值递增有序排列。试对顺序表编写求C的算法。