

试卷代号:1010

座位号

中央广播电视大学 2008—2009 学年度第一学期“开放本科”期末考试

### 数据结构 试题

2009 年 1 月

题号	一	二	三	四	五	六	总分
分数							

得分	评卷人

一、单项选择题(在括号内填写所选择的标号。每小题 2 分,共 18 分)

得分  1. 下面算法的时间复杂度为( )。

```
int f(unsigned int n) {
    if (n==0 || n==1) return 1;
    else return n * f(n-1);
}
```

- A.  $O(1)$
- B.  $O(n)$
- C.  $O(n^2)$
- D.  $O(n!)$

得分  2. 在一个长度为  $n$  的线性表中顺序查找一个值为  $x$  的元素时,在等概率的情况下,查找成功时的平均查找长度为( )。

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

得分  3. 已知  $L$  为一个单链表的表头指针,在表头插入结点 \*  $p$  的操作是( )。

- A.  $p=L; p->link=L;$
- B.  $p->link=L; p=L;$
- C.  $p->link=L; L=p;$
- D.  $L=p; p->link=L;$

得分  4. 若有一个循环队列  $Q$ ,队首和队尾指针分别为  $front$  的  $rear$ ,则判断队列满的条件为( )。

- A.  $Q.front==Q.rear;$
- B.  $Q.front-Q.rear==MaxSize;$
- C.  $Q.front+Q.rear==MaxSize;$
- D.  $Q.front==(Q.rear+1)\%MaxSize;$

得分  5. 在一棵完全二叉树中,若编号为  $i$  的结点存在左子女,则左子女结点的编号为( ),假定树根结点的编号为 0。

- A.  $2i$
- B.  $2i-1$
- C.  $2i+1$
- D.  $2i+2$

- 得分  6. 对长度为 10 的线性有序表  $A[10]$  进行折半查找, 若查找到元素  $A[2]$  时成功, 则查找长度为( )。
- A. 1                                    B. 2  
C. 3                                    D. 4
- 得分  7. 向一棵 AVL 树插入元素时, 可能引起对最小不平衡子树的调整过程, 此调整过程共具有( )种旋转类型。
- A. 1                                    B. 2  
C. 3                                    D. 4
- 得分  8. 设一个有向图具有  $n$  个顶点和  $e$  条边, 若采用邻接表作为其存储结构, 则空间复杂度为( )。
- A.  $O(n \times \log_2 e)$                     B.  $O(n+e)$   
C.  $O(n^e)$                             D.  $O(n^2)$
- 得分  9. 在一棵  $m$  阶 B 树中, 树根结点所包含的关键码个数至少为( )。
- A. 1                                    B. 2  
C.  $m/2$                                 D.  $m-1$

得 分	评卷人

二、填空题(在横线处填写合适的内容。每小题 2 分, 共 14 分)

- 得分  10. 利用三元组表存放稀疏矩阵中的非零元素, 则在三元组表中每个三元组元素对应一个非零元素的行号、列号和\_\_\_\_\_。
- 得分  11. 在单链表中逻辑上相邻的结点而在物理位置上\_\_\_\_\_相邻。
- 得分  12. 在一棵高度为 2 的四叉树中, 最多含有\_\_\_\_\_个结点, 假定树根结点的高度为 0。
- 得分  13. 在一个堆的顺序存储中, 若一个元素的下标为  $i(i \geq 0)$ , 则它的右子女元素的下标为\_\_\_\_\_。
- 得分  14. 根据一组记录(56, 42, 73, 50, 48, 22)依次插入结点生成一棵 AVL 树时, 当插入到值为\_\_\_\_\_的结点时首次出现不平衡, 需要进行旋转调整。
- 得分  15. 当利用 Kruskal 算法求解一个连通网的最小生成树时, 只有当一条候选边的两个端点不在同一个\_\_\_\_\_分量时, 才会被加入到最小生成树中。
- 得分  16. 在堆排序中, 对  $n$  个记录建立初始堆需要调用\_\_\_\_\_次筛运算的算法。

得分	评卷人

三、判断题(在每小题后面括号内打对号表示叙述正确或打叉号表示叙述错误。每小题 2 分,共 14 分)

得分  17. 线性表若采用链式存储表示时,其存储结点的地址可连续也可不连续。 ( )

得分  18. 在用单链表表示的链式队列 Q 中,假定队头指针为  $Q \rightarrow front$ ,队尾指针为  $Q \rightarrow rear$ ,则链队为空的条件为  $Q \rightarrow front == Q \rightarrow rear$ 。 ( )

得分  19. 一棵 AVL 树的所有叶结点不一定在同一层上。 ( )

得分  20. 在一棵二叉树中,假定每个结点只有左子女,没有右子女,若对它分别进行中序遍历和后序遍历,则具有相同的遍历结果。 ( )

得分  21. 折半搜索所对应的判定树,既是一棵二叉搜索树,又是一棵理想平衡二叉树。 ( )

得分  22. 对一个用顶点表示活动的网络(AOV 网)进行拓扑排序的结果是唯一的。 ( )

得分  23. 堆排序是一种稳定的排序方法。 ( )

得分	评卷人

四、计算题(每小题 6 分,共 30 分)

得分  24. 假定一棵普通树的广义表表示为  $a(b(e), c(f(h, i, j), g))$ ,分别写出对其进行先根、后根和按层遍历的结果。

先根:

后根:

按层:

得分  25. 假定一个线性表为  $(38, 52, 25, 74, 68, 16, 30)$ ,根据此表中的元素排列次序生成一棵二叉搜索树,求出该二叉搜索树中元素为 25、74、68 结点的层号。假定树根结点的层号为 1。

结点:

25	74	68

层号:

得分	
----	--

26. 已知一个数据集合为{28,12,16,49,34,30},试把它调整为一个最大堆。

最大堆:

得分	
----	--

27. 已知一个图的顶点集 V 和边集 G 分别为:

$V = \{1, 2, 3, 4, 5\};$

$E = \{ \langle 1, 2 \rangle, \langle 1, 3 \rangle, \langle 2, 4 \rangle, \langle 2, 5 \rangle, \langle 3, 4 \rangle, \langle 4, 5 \rangle, \langle 5, 1 \rangle, \langle 5, 3 \rangle \};$

假定此图采用邻接矩阵表表示,根据图的遍历算法分别写出从顶点 1 出发进行深度优先搜索和广度优先搜索所得到的顶点序列。

深度优先搜索序列:

广度优先搜索序列:

得分	
----	--

28. 设散列表的长度  $m=7$ ;散列函数为  $H(K) = K \bmod m$ ,若采用线性探查法解决冲突,待依次插入的关键码序列为{19,14,23,68,20},分别求出查找 23、68、20 时的搜索长度。

查找 23、68、20 的搜索长度:

得 分	评卷人

### 五、计算分析题(每小题 6 分,共 12 分)

得分	
----	--

29. 设 rear 是以循环链表表示的队列的队尾指针,EnLQueue 函数实现把 x 插入到队尾的操作。阅读算法,在划有横线的上面填写合适的内容。

```
void EnLQueue(ListNode * & rear, ElemType x)
{
    ListNode * p;
    p=new ListNode;          //p 指向动态分配的结点空间
    p->data=x;
    p->link=_____ ;
    rear->link=p;_____ ;
};
```

得分	
----	--

30. 假定 HL 为一个单链表的表头指针, K 为一个待查找的值, 请指出算法功能。

```
ListNode * Unknown(ListNode * HL, int K)
{
    if(HL==NULL) return NULL;
    if(HL->data==K) return HL;
    ListNode * cp;
    cp=HL->link;
    while(cp!=NULL)
        if(cp->data==K) return cp;
        else cp=cp->link;
    return NULL;
}
```

算法功能:

得分	评卷人

### 六、编写题(每小题 6 分, 共 12 分)

得分	
----	--

31. 试编写一个函数, 在一个顺序表 A 中顺序查找出元素的最大值, 由引用参数

Max 带回。函数的原型如下:

```
#include "SeqList. h"
template <class T>
void FindMaxMin(SeqList<int>& A, int& Max);
```

在编写此函数时可以使用顺序表类中定义的如下两个公有函数:

```
int Length(); //求表的长度;
int getData(int k); //提取第 k 个元素的值, 0≤k<A.Length()。
```

```
void FindMaxMin(SeqList<int>& A, int& Max) //向下写出函
```

数体

得分

32. 已知二叉树中的结点类型 BinTreeNode 定义为：

```
struct BinTreeNode {char data; BinTreeNode * left, * right;};
```

其中 data 为结点值域, left 和 right 分别为指向左、右子女结点的指针域。根据下面函数声明编写出判断两棵二叉树是否相等的算法, 若相等则返回 1 否则返回 0。算法中参数 T1 和 T2 分别指向这两棵二叉树的根结点。当两棵树的结结构完全相同并且对应结点的值也相同时才被认为相等。

```
int BTreeEqual(BinTreeNode * T1, BinTreeNode * T2);
```

试卷代号:1010

中央广播电视大学 2008—2009 学年度第一学期“开放本科”期末考试

数据结构 试题答案及评分标准

(供参考)

2009 年 1 月

一、单项选择题(在括号内填写所选择的标号。每小题 2 分,共 18 分)

1. B            2. C            3. C            4. D            5. C  
6. C            7. D            8. B            9. A

二、填空题(在横线处填写合适的内容。每小题 2 分,共 14 分)

10. 值  
11. 不一定  
12. 21  
13.  $2i+2$   
14. 48  
15. 连通  
16.  $n/2$

三、判断题(在每小题后面括号内打对号表示叙述正确或打叉号表示叙述错误。每小题 2 分,共 14 分)

17. ✓            18. ✗            19. ✓            20. ✓            21. ✓  
22. ✗            23. ✗

四、计算题(每小题 6 分,共 30 分)

24. 先根:a,b,e,c,f,h,i,j,g //2 分  
后根:e,b,h,i,j,f,g,c,a //2 分  
按层:a,b,c,e,f,g,h,i,j //2 分

25. 评分标准:每个数值对得 2 分,全对给 6 分

结点:

25	74	68
----	----	----

层号:

2	3	4
---	---	---

26. {49,34,30,12,28,16}

27. 深度搜索序列:1,2,4,5,3 //3 分

广度搜索序列:1,2,3,4,5 //3 分

28. 查找 23、68、20 的搜索长度:1、2、3 //每个数据占 2 分

### 五、计算分析题(每小题 6 分,共 12 分)

29. rear->link, rear=p //每空 3 分

30. 从 HL 单链表中顺序查找出值为 K 的结点,若查找成功则返回该结点的地址,否则返回空。

### 六、编写题(每小题 6 分,共 12 分)

31. 请根据编程的完整程度酌情给分。

```
#include "SeqList.h"
template <class T>
void FindMaxMin(SeqList<int>& A, int& Max)
{
    Max=A.getData(0); //1 分
    for(int i=1; i<A.Length();i++) //3 分
        if(A.getData(i)>Max) Max=A.getData(i); //6 分
}
```

32. 请根据编程的完整程度酌情给分

```
int BTreeEqual(BinTreeNode * T1,BinTreeNode * T2)
{
    //若两棵树均为空则返回 1 表示相等
    if(T1==NULL && T2==NULL) return 1; //1 分

    //若一棵为空一棵不为空则返回 0 表示不等
```



```

    else if(T1==NULL || T2==NULL) return 0; //2分
//若根结点值相等并且左、右子树对应相等则两棵树相等
    else if(T1->data==T2->data && BTreeEqual(T1->left, T2->left) &&
        BTreeEqual(T1->right, T2->right)) return 1; //5分
//若根结点值不等或左、右子树对应不等则两棵树不等
    else return 0; //6分
}

```

另一个参考答案:

```

int BTreeEqual(BinTreeNode * T1, BinTreeNode * T2)
{
//若两棵树均为空或实际上是同一棵树时返回 1 表示相等
    if(T1==T2) return 1; //1分
//若一棵为空一棵不为空则返回 0 表示不等
    if(T1==NULL || T2==NULL) return 0; //2分
//若根结点值不等返回 0 表示不等
    if(T1->data!=T2->data) return 0; //3分
//若根结点值相等则两棵树是否相等取决于它们的左、右子树是否对应相等
    return BTreeEqual(T1->left, T2->left) && BTreeEqual(T1->right, T2->right);
//6分
}

```