

试卷代号:1010

座位号

中央广播电视大学 2006—2007 学年度第二学期“开放本科”期末考试

### 计算机专业 数据结构 试题

2007 年 7 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)

- 若需要利用形参直接访问实参,则应把形参变量说明为( )参数。  
 A. 指针  
 B. 引用  
 C. 传值  
 D. 常值
- 在二维数组中,每个数组元素同时处于( )个向量中。  
 A. 0  
 B. 1  
 C. 2  
 D. n
- 已知单链表 A 长度为 m,单链表 B 长度为 n,它们分别由表头指针所指向,若将 B 整体连接到 A 的末尾,其时间复杂度应为( )。  
 A.  $O(1)$   
 B.  $O(m)$   
 C.  $O(n)$   
 D.  $O(m+n)$
- 假定一个链式队列的队头和队尾指针分别为 front 和 rear,则判断队空的条件为( )。  
 A.  $front == rear$   
 B.  $front != NULL$   
 C.  $rear != NULL$   
 D.  $front == NULL$
- 若让元素 1,2,3 依次进栈,则出栈次序不可能出现( )种情况。  
 A. 3,2,1  
 B. 2,1,3  
 C. 3,1,2  
 D. 1,3,2

6. 在一棵高度为 5(假定树根结点的高度为 0)的完全二叉树中,所含结点个数至少等于 ( )

A. 16

B. 64

C. 31

D. 32

7. 向具有  $n$  个结点的二叉搜索树中插入一个结点的时间复杂度大致为( )。

A.  $O(1)$

B.  $O(\log_2 n)$

C.  $O(n)$

D.  $O(n\log_2 n)$

8. 具有  $n$  个顶点的有向图最多可包含有( )条有向边。

A.  $n-1$

B.  $n$

C.  $n(n-1)/2$

D.  $n(n-1)$

9. 图的广度优先搜索类似于树的( )遍历。

A. 先根

B. 中根

C. 后根

D. 层次

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)

1. 链表只适用于\_\_\_\_\_查找。

2. 设双向循环链表中每个结点的结构为(data, llink, rlink),则结点 \* p 的前驱结点的地址为\_\_\_\_\_。

3. 在一个链式队列中,若队头指针与队尾指针的值相同,则表示该队列至多有\_\_\_\_\_个结点。

4. 假定一棵树的广义表表示为 a(b, c, d(e, f), g(h)),则结点 f 的层数为\_\_\_\_\_。假定树根结点的层数为 0。

5. 从一棵二叉搜索树中搜索一个元素时,若给定值大于根结点的值,则需要向根的\_\_\_\_\_继续搜索。

6. 每次从第  $i$  至第  $n$  个元素中顺序挑选出一个最小元素,把它交换到第  $i$  个位置,此种排序方法叫做\_\_\_\_\_排序。

7. 快速排序在最坏情况下的时间复杂度为\_\_\_\_\_。

得分	评卷人

三、判断题,在每小题前面打对号表示正确或打叉号表示错误(每小题 2 分,共 14 分)

- ( ) 1. 数据的逻辑结构与数据元素本身的内容和形式无关。
- ( ) 2. 使用三元组表示稀疏矩阵中的非零元素能节省存储空间。
- ( ) 3. 在一棵二叉树中,假定每个结点只有左子女,没有右子女,则对它分别进行前序遍历和按层遍历时具有相同的结果。
- ( ) 4. 能够在链接存储的有序表上进行折半搜索,其时间复杂度与在顺序存储的有序表上相同。
- ( ) 5. 邻接表表示只能用于有向图的存储,邻接矩阵对于有向图和无向图的存储都适用。
- ( ) 6. 在索引顺序结构上实施分块搜索,在等概率情况下,其平均搜索长度不仅与子表个数有关,而且与每一个子表中的对象个数有关。
- ( ) 7. 向一棵 B 树插入关键码的过程中,若最终引起树根结点的分裂,则新树比原树的高度减少 1。

得分	评卷人

四、运算题(每小题 6 分,共 30 分)

1. 假定一棵二叉树广义表表示为  $a(b(c(g)),d(e,f))$ ,分别写出对它进行先序、中序和后序遍历的结果。

先序:

中序:

后序:

2. 有 7 个带权结点,其权值分别为 3,7,8,2,6,10,14,试以它们为叶子结点生成一棵霍夫曼树,求出该树的带权路径长度。

带权路径长度:

3. 已知图  $G=(V,E)$ ,其中

$$V=\{a,b,c,d,e\},$$

$$E=\{\langle a,b\rangle,\langle b,a\rangle,\langle c,b\rangle,\langle c,d\rangle,\langle d,e\rangle,\langle e,a\rangle,\langle e,c\rangle\}$$

在该图的邻接表表示中,每个顶点单链表各有多少个边结点。

顶点:            a            b            c            d            e

边结点数:

4. 已知一个 AOV 网的顶点集  $V$  和边集  $G$  分别为:

$V = \{0, 1, 2, 3, 4, 5, 6, 7\}$ ;

$E = \{\langle 0, 2 \rangle, \langle 1, 3 \rangle, \langle 1, 4 \rangle, \langle 2, 4 \rangle, \langle 2, 5 \rangle, \langle 3, 6 \rangle, \langle 3, 7 \rangle, \langle 4, 7 \rangle, \langle 5, 7 \rangle, \langle 6, 7 \rangle\}$ ;

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,则按主教材中介绍的进行拓扑排序的算法,写出得到的拓扑序列。

拓扑序列:

5. 已知有一个数据表为  $\{30, 16, 20, 15, 38, 12, 44, 53, 46, 18, 26, 86\}$ , 给出进行归并排序的过程中每一趟排序后的数据表变化。

(0) [30 16 20 15 38 12 44 53 46 18 26 86]

(1)

(2)

(3)

(4)

得分	评卷人

### 五、算法分析题(每小题 6 分,共 12 分)

1. 该算法功能为:从表头指针为  $la$  的、按值从小到大排列的有序链表中删除所有值相同的多余元素,并释放被删结点的动态存储空间。阅读算法,在划有横线的上面填写合适的内容。

```
void purge_linkst(ListNode * & la)
{
    ListNode * p, * q;
    if(la == NULL) return;
    q = la; p = la->link;
    while(p) {
        if(p->data > q->data) {q = p; p = p->link;}
        else {
            q->link = _____;
            delete(p);
            p = _____;
        }
    }
}
```

2. 已知二叉树中的结点类型 BinTreeNode 定义为:

```
struct BinTreeNode {ElemType 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。下面函数的功能是从二叉树 BT 中查找值为 X 的结点, 若查找成功则返回结点地址, 否则返回空。阅读算法, 在划有横线的上面填写合适的内容。

```
BinTreeNode * BTF(BinTreeNode * BT, ElemType x)
{
    if(BT==NULL) NULL;
    else {
        if(BT->data==x) _____;
        else {
            BinTreeNode * t;
            if(t=BTF(BT->left, x)) return t;
            if(t=_____ ) return t;
            return NULL;
        }
    }
}
```

得分	评卷人

### 六、算法设计题(每小题 6 分, 共 12 分)

1. Q 是一个由其队尾指针和队列长度标识的循环队列, 请写出插入一个元素的算法。

```
struct CyclicQueue // 循环队列定义
{
    ElemType elem[M]; //M 为已定义过的整型常量
    int rear; // rear 指向队尾元素的后一个位置
    int length; // length 指示队列中元素个数
};
```

```
bool EnCQueue(CyclicQueue& Q, ElemType x);
//Q 是一个循环队列, 若队列不满, 则将 x 插入并返回 true; 否则返回 false。
//在下面写出该函数的函数体
```

2. 已知二叉搜索树中的结点类型 BinTreeNode 定义为:

```
struct BinTreeNode {ElemType 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。参数 BST 指向一棵二叉搜索树的根结点。试根据下面的函数声明编写一个递归算法, 向 BST 树中插入值为 item 的结点, 要求若树中不存在 item 结点则进行插入并返回 1 表示插入成功, 若树中已存在 item 结点则不插入并返回 0 表示插入失败。

```
int Insert(BinTreeNode * & BST, const ElemType& item);
```

试卷代号:1010

中央广播电视大学 2006—2007 学年度第二学期“开放本科”期末考试

计算机专业 数据结构 试题答案及评分标准

(供参考)

2007 年 7 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. B            4. D            5. C  
6. D            7. B            8. D            9. D

二、填空题,在横线处填写合适内容(每小题 2 分,共 14 分)

1. 顺序            2.  $p \rightarrow llink$             3. 1            4. 2  
5. 右子树            6. 直接选择            7.  $O(n^2)$

三、判断题,在每小题前面打对号表示正确或打叉号表示错误(每小题 2 分,共 14 分)

1. 对            2. 对            3. 对            4. 错  
5. 错            6. 对            7. 错

四、运算题(每小题 6 分,共 30 分)

1. 先序:a,b,c,g,d,e,f //2 分  
中序:c,g,b,a,e,d,f //2 分  
后序:g,c,b,e,f,d,a //2 分

2. 带权路径长度:131

3. 评分标准:每个数据对给 1 分,全对给 6 分。

顶点:            a            b            c            d            e  
边结点数:        1            1            2            1            2

4. 评分标准:若与答案完全相同得 6 分,若仍为一种拓扑序列则得 3 分,其他则酌情处理。

拓扑序列:1,3,6,0,2,5,4,7

5. 分步给分

- (0) [30 16 20 15 38 12 44 53 46 18 26 86]  
(1) [16 30][15 20][12 38][44 53][18 46][26 86] //1 分  
(2) [15 16 20 30][12 38 44 53][18 26 46 86] //1 分  
(3) [12 15 16 20 30 38 44 53][18 26 46 86] //2 分  
(4) [12 15 16 18 20 26 30 38 44 46 53 86] //2 分

五、算法分析题(每小题 6 分,共 12 分)

1. `p->link,q->link` //每空 3 分
2. `return BT、BTF(BT->right, x)` //每空 3 分

六、算法设计题(每小题 6 分,共 12 分)

评分标准:根据编程完整程度酌情给分。

1. `bool EnCQueue(CyclicQueue& Q, ElemType x);`  
{  
    if(Q.length==M) return false; //1 分  
    Q.elem[Q.rear]=x; //2 分  
    Q.rear=(Q.rear+1)%M; //4 分  
    Q.length++; //5 分  
    return true; //6 分  
}
2. `int Insert(BinTreeNode * & BST, const ElemType& item)`  
{  
    if(BST==NULL){  
        BinTreeNode \* p=new BinTreeNode;  
        p->data=item;  
        p->left=p->right=NULL;  
        BST=p;  
        return 1;  
    } //3 分  
    else if(item==BST->data) return 0; //4 分  
    else if(item<BST->data)  
        return Insert(BST->left, item); //5 分  
    else  
        return Insert(BST->right, item); //6 分  
}

说明:函数体中的三个 else 保留字均可以被省略。