

试卷代号:1010

座位号

中央广播电视大学 2002—2003 学年度第一学期“开放本科”期末考试

### 计算机专业数据结构试题

2003 年 1 月

题 号	一	二	三	四	五	六	总 分
分 数							

得 分	评卷人

#### 一、选择题(每小题 1 分,共 10 分)

得分  1. 在一个长度为  $n$  的顺序表的表尾插入一个新元素的渐进时间复杂度为( )。

- A.  $O(n)$
- B.  $O(1)$
- C.  $O(n^2)$
- D.  $O(\log_2 n)$

得分  2. 设单链表中结点的结构为(data, link)。已知指针  $q$  所指结点是指针  $p$  所指结点的直接前驱,若在  $*q$  与  $*p$  之间插入结点  $*s$ ,则应执行下列哪一个操作? ( )

点的直接前驱,若在  $*q$  与  $*p$  之间插入结点  $*s$ ,则应执行下列哪一个操作? ( )

- A.  $s \rightarrow \text{link} = p \rightarrow \text{link}; p \rightarrow \text{link} = s$
- B.  $q \rightarrow \text{link} = s; s \rightarrow \text{link} = p$
- C.  $p \rightarrow \text{link} = s \rightarrow \text{link}; s \rightarrow \text{link} = p$
- D.  $p \rightarrow \text{link} = s; s \rightarrow \text{link} = q$

得分  3. 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

得分  4. 一个递归的定义可以用递归过程求解,也可以用非递归过程求解,但单从运行时间来看,通常递归过程比非递归过程( )。

- A. 较快
- B. 较慢
- C. 相同

得分  5. 树中所有结点的度等于所有结点数加( )。

- A. 0
- B. 1
- C. -1
- D. 2

得分  6. 在一棵具有  $n$  个结点的二叉树中,所有结点的空子树个数等于( )。

- A.  $n$
- B.  $n-1$
- C.  $n+1$
- D.  $2 * n$

得分  7. 对长度为  $n$  的有序单链表,若搜索每个元素的概率相等,则顺序搜索到表中任一元素的平均搜索长度为( )。

- A.  $n/2$
- B.  $(n+1)/2$
- C.  $(n-1)/2$
- D.  $n/4$

得分  8. 在无向图中定义顶点  $v_i$  与  $v_j$  之间的路径为从  $v_i$  到达  $v_j$  的一个( )。

- A. 顶点序列
- B. 边序列
- C. 权值总和
- D. 边的条数

得分  9. 如果只想得到 1024 个元素组成的序列中的前 5 个最小元素,那么用( )方法最快。

- A. 起泡排序
- B. 快速排序
- C. 堆排序
- D. 直接选择排序

得分  10. 设有一个含 200 个表项的散列表,用线性探查法解决冲突,按关键码查询时找到一个表项的平均探查次数不超过 1.5,则散列表项应能够至少容纳( )个表项。

(设搜索成功的平均搜索长度为  $S_{s1} = \{1 + 1/(1-\alpha)\}/2$ ,其中  $\alpha$  为装填因子)

- A. 400
- B. 526
- C. 624
- D. 676

得分	评卷人

## 二、填空题(每小题 1 分,共 10 分)

得分  11. 在程序运行过程中不能扩充的数组是\_\_\_\_\_分配的数组。这种数组在声明它时必须指定它的大小。

得分  12. 将一个  $n$  阶三对角矩阵  $A$  的三条对角线上的元素按行压缩存放于一个一维数组  $B$  中,  $A[0][0]$  存放于  $B[0]$  中。对于任意给定数组元素  $A[I][J]$ , 如果它能够在数组  $B$  中找到, 则它应在\_\_\_\_\_位置。

得分  13. 链表适用于\_\_\_\_\_查找。

得分  14. 队列的插入操作在\_\_\_\_\_进行, 删除操作在\_\_\_\_\_进行。

得分  15. 设有一个顺序栈  $S$ , 元素  $s_1, s_2, s_3, s_4, s_5, s_6$  依次进栈, 如果 6 个元素的出栈顺序为  $s_2, s_3, s_4, s_6, s_5, s_1$ , 则顺序栈的容量至少应为\_\_\_\_\_。

得分  16. 通常程序在调用另一个程序时, 都需要使用一个\_\_\_\_\_来保存被调用程序内分配的局部变量、形式参数的存储空间以及返回地址。

得分  17. 广义表  $A((a, b, c), (d, e, f))$  的表尾为\_\_\_\_\_。

得分  18. 在一棵树中, \_\_\_\_\_结点没有前驱结点。

得分  19. 一棵树的广义表表示为  $a(b(c,d(e,f),g(h)),i(j,k(x,y)))$ , 结点  $k$  的所有祖先的结点数为 \_\_\_\_\_ 个。

得分  20. 根据一组记录(56,42,50,64,48)依次插入结点生成一棵 AVL 树(高度平衡的二叉搜索树)时,当插入到值为 \_\_\_\_\_ 的结点时需要进行旋转调整。

得分	评卷人
<input type="text"/>	<input type="text"/>

三、判断题(每小题 1 分,共 10 分)

得分  21. 二维数组可以视为数组元素为一维数组的一维数组。 ( )

得分  22. 链接存储表示的存储空间一般在程序的运行过程中动态分配和释放,通常存储器中还有空闲存储空间,就不会产生存储溢出的问题。 ( )

得分  23. 在用单链表表示的链式队列中,队头在链表的链尾位置。 ( )

得分  24. 凡是递归定义的数据结构都可以用递归算法来实现它的操作。 ( )

得分  25. 当向一个小根堆(最小堆)中插入一个具有最小值的元素时,该元素需要逐层向上调整,直到被调整到堆顶位置为止。 ( )

得分  26. 对于一棵具有  $n$  个结点,其高度为  $h$  的二叉树,进行任一种次序遍历的时间复杂度为  $O(n)$ 。 ( )

得分  27. 进行折半搜索的表必须是顺序存储的有序表。 ( )

得分  28. 一棵  $m$  阶 B 树中每个结点最多有  $m$  个关键码,最少有 2 个关键码。 ( )

得分  29. 在散列法中采取开散列(链地址)法来解决冲突时,其装载因子的取值一定在  $(0,1)$  之间。 ( )

得分  30. 一棵 3 阶 B 树是平衡的 3 路搜索树,反之,一棵平衡的 3 路搜索树是 3 阶 B 树。 ( )

得分	评卷人

#### 四、运算题(每小题 8 分,共 40 分)

得分  31. 对于一个  $n \times n$  的矩阵 A 的任一矩阵元素  $a[i][j]$ , 按行存储时和按列存储时的地址之差是多少。(若设两种存储的开始存储地址  $LOC(0,0)$  及元素所占存储单元数  $d$  相同)

得分  32. 设有一个求解汉诺塔(Hanoi)的递归算法

```
void HANOI(int n, int peg1, int peg2, int peg3){
    if (n==1) cout <<"move " <<peg1<<" to " <<peg3<<endl;
    else {HANOI (n-1, peg1, peg3, peg2);
        cout<<"move " <<peg1<<" to " <<peg3<<endl;
        HANOI(n-1, peg2, peg1,peg3);
    }
}
```

假定采用  $HANOI(3,1,2,3)$  去调用上述算法, 则写出整个输出结果的前四行内容。

得分  33. 已知一棵二叉树的中序和前序序列如下, 求该二叉树的后序序列。

中序序列: c, b, d, e, a, g, i, h, j, f

前序序列: a, b, c, d, e, f, g, h, i, j

后序序列:

得分  34. 假定一组数据对象为(40,28,16,56,50,32,30,63), 按次序插入每个对象生成一棵 AVL 树(高度平衡的二叉搜索树), 根据插入过程填写下表, 在相应位置填写所需要的调整类型: “左单旋转”、“右单旋转”、“先左后右双旋转”、“先右后左双旋转”, 若不需要旋转则填写“无”。

数据:	40	28	16	56	50	32	30	63
调整:								

得分  35. 设有 150 个表项要存储到散列表中,要求利用线性探查法解决冲突,同时要求找到所需表项的平均比较次数不超过 2 次。试问散列表 m 至少需要设计多大? 设  $\alpha$  是散列表的装载因子,则有

$$ASL_{succ} = \frac{1}{2} \left( 1 + \frac{1}{1-\alpha} \right)$$

$$m \geq \underline{\hspace{2cm}}$$

得分	评卷人

### 五、算法分析题(每小题 8 分,共 24 分)

得分  36. 设顺序表 SeqList 具有下列操作:

```
int Length() const;    //计算表长度
T Remove();           //删除当前表项,置下一表项为当前表项
T First();             //取表中第一个表项的值,并把它置为当前表项
```

项

```
T Next();             //取当前表项的后继表项的值,并把该后继表项置为当前表项
```

前表项

试针对下面所给算法,回答下列问题:

(1)说明若顺序表中存放的数据为{29,38,47,16,95,64,73,83,51,10,0,26},表的长度为 12,参数值  $s=10, t=30$ ,写出执行算法后得到的顺序表。

(2)说明该算法的功能。

(3)若表长用  $n$  表示,说明它的渐进时间复杂度。

```
#include <iostream.h>
```

```
#include "SeqList.h"
```

```
template <T>
```

```
void unknown (SeqList<T> * L, T s, T t){
```

```
if(! L->Length() || s>=t)
```

```
{cerr<<"List is empty or parameters are illegal!"<<endl; exit(1);}
```

```
int i=0;
```

```
T temp =L->First();
```

```
while(i<L->Length())
```

```
if(temp >=s && temp <=t) L->Remove();
```

```
else { temp=L->Next(); i++; }
```

```
}
```

得分

37. 将二项式 $(a+b)^i$ 展开,其系数构成杨辉三角形。若想按行将展开式系数的前 $n$ 行打印出来,需要用到一个队列,存放各行展开式的系数。假定 $n=3$ ,试根据下列算法写出输出结果。

```
#include "queue. h"
void YANGVI (int n){
    Queue q(n+2);
    q. EnQueue(1); q. EnQueue(1);
    int i, j , t, s =0;
    for (i=1; i<=n; i++){
        cout <<endl;
        q. EnQueue (0);
        for (j=1; j<=i+2; j++){
            q. DeQueue (t);
            q. EnQueue (s+t);
            s=t;
            if(j! =i+2)cout<<s<<' ';
        }
    }
}
```

输出结果为:

得分

38. 已知二叉树中的结点类型用 BinTreeNode 表示,被定义为:

```
struct BinTreeNode{ElemType data; BinTreeNode * leftChild, * rightChild};
```

其中 data 为结点值域, leftChild 和 rightChild 分别为指向左、右子女结点的指针域。下面函数的功能是返回二叉树 BT 中值为 x 的结点所在的层号,请在划有横线的地方填写合适的内容。

```
int NodeLevel(BinTreeNode * BT, ElemType&x){
    if (BT==NULL)return -1; //空树的层号为-1
    else if (BT->data==x)return 0; //根结点的层号为0
    else{
        int c1=NodeLevel(BT->leftChild,x); //向左子树中查找 x 结点
        if(c1>=0) _____;
        int c2= _____;
        if(c2>=0) _____;
        else return -1; //在树中不存在 x 结点返回-1
    }
}
```

得 分	评卷人

六、算法设计题(6分)

39. 已知二叉树中的结点类型用 BinTreeNode 表示,被定义为:

struct BinTreeNode { char data; BinTreeNode \* leftChild, \* rightChild;};其中 data 为结点值域, leftChild 和 rightChild 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求一棵二叉树高度的算法,该高度由函数返回。假定根结点的层次为 0,参数 BT 初始指向这棵二叉树的根结点。

```
int BTreeHeight (BinTreeNode * BT);
```



试卷代号:1010

中央广播电视大学 2002—2003 学年度第一学期“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)

2003 年 1 月

一、选择题(每小题 1 分,共 10 分)

- |      |      |      |      |       |
|------|------|------|------|-------|
| 1. B | 2. B | 3. C | 4. B | 5. C  |
| 6. C | 7. B | 8. A | 9. D | 10. A |

二、填空题(每小题 1 分,共 10 分)

11. 静态
12.  $2 * I + J$
13. 顺序
14. 队尾      队头
15. 3
16. 栈
17. ((d,e,f))
18. 根
19. 2
20. 50

三、判断题(每小题 1 分,共 10 分)

21. 对
22. 对
23. 错
24. 对
25. 对

26. 对  
 27. 对  
 28. 错  
 29. 错  
 30. 错

四、运算题(每题 8 分,共 40 分)

31. 按行存储时与按列存储时,计算  $A[i][j]$  地址的公式分别为

$$LOC(i,j) = LOC(0,0) + (i * n + j) * d$$

及  $LOC'(i,j) = LOC(0,0) + (j * n + i) * d$

两者相减,得

$$\begin{aligned} LOC(i,j) - LOC'(i,j) &= LOC(0,0) + (i * n + j) * d - LOC(0,0) - (j * n + i) * d \\ &= (i-j) * n * d + (j-i) * d \text{ 或:} \\ &= (i-j) * (n-1) * d \end{aligned}$$

32. 汉诺塔的递归处理过程如下表示:

- move 1 to 3 (2分)  
 move 1 to 2 (2分)  
 move 3 to 2 (2分)  
 move 1 to 3 (2分)

33. 后序序列:c,e,d,b,i,j,h,g,f,a

34. 插入结果和调整类型为

插入数据:	40	28	16	56	50	32	30	63
调整类型:	无	无	右单旋	无	先右后左 双旋转	先右后左 双旋转	无	左单旋

//每一个元素调整类型正确给 1 分。

35. 已知要存储的表项数为  $n=150$ , 搜索成功的平均搜索长度为  $ASL_{succ} \leq 2$ , 则有  $ASL_{succ}$

$$= \frac{1}{2} \left( 1 + \frac{1}{1-\alpha} \right) \leq 2, \text{ 解得 } \alpha \leq \frac{2}{3}. \text{ 又有 } \alpha = \frac{n}{m} = \frac{150}{m} \leq \frac{2}{3}, \text{ 则 } m \geq 225.$$

五、算法分析题(每小题 8 分,共 24 分)

36. (1)算法执行后得到的顺序表为{38,47,95,64,73,83,51,0},表的长度减为 8。(4 分)

(2)该算法的功能是在顺序表中删除其值在给定值  $s$  与  $t$  之间(要求  $s$  小于  $t$ )的所有元素。(2 分)

(3)此算法的渐进时间复杂度为  $O(n)$ 。(2 分)

37. 1 1 //3 分

1 2 1 //3 分

1 3 3 1 //2 分

38. (1)**return** c1+1 //3 分

(2)NodeLevel(BT->rightChild,X) //3 分

(3)**return** c2+1 //2 分

六、6 分,请根据编程情况酌情给分。

39. 算法如下

```
int BTreeHeight(BinTreeNode * BT){
    if (BT == NULL) return -1; //对于空树,返回-1 并结束递归,1 分
    else {
        int h1=BTreeHeight(BT->leftChild); //计算左子树的高度,2 分
        int h2=BTreeHeight(BT->rightChild); //计算右子树的高度,2 分
        if(h1>h2) return h1+1;
        else return h2+1; //返回树的高度,1 分
    }
}
```