

2021 天勤计算机考研八套模拟卷 · 卷一

数据结构篇

一、选择题（单选）

1. 利用栈求下列表达式的值（一个操作数栈，一个运算符栈，并假设每个栈只有两个存储单元），则在下列表达式中，不会发生溢出的是（ ）。

- A. $(A-B)*(C-D)$
- B. $(A-B*C)-D$
- C. $(A-B)*C-D$
- D. $A-B*(C-D)$

2. 某栈最多容纳 3 个元素，入栈的序列为 1、2、3、4、5 则出栈序列可能为（ ）。

- A. 4、3、2、1、5
- B. 1、5、4、3、2
- C. 5、4、3、2、1
- D. 3、2、1、5、4

3. 某平衡二叉树的树高为 3，其根结点 A 左孩子的平衡因子为 -1，右孩子的度为 0。在该平衡二叉树中插入一个结点后造成了不平衡，则应该进行（ ）型旋转以使其平衡。

- A. LR 或者 LL B. LL 或者 RL
- C. RR 或者 LL D. RL 或者 RR

4. 将 5 个字母“oops”按此顺序入栈，则有（ ）种不同的出栈顺序可以仍然得到“oops”。

- A. 1 B. 3 C. 5 D. 6

5. 假设 n 是描述问题规模的非负整数，下面程序片段的时间复杂度为（ ）。

- A. $O(n^2 \log_2 n)$ B. $O(n \log_5 n)$
- C. $O(n^2 \log_5 n)$ D. $O(n^3)$

```
void fun(int n){
    int i,j,k;
    for(i=1;i<=n;i++)
        for(j=1;j<=n;j++){
            k=1;
            while(k<=n)
                k=5*k;
        }
}
```

6. 一棵三叉树中，已知度为 3 的结点个数等于度为 2 的结点数，且树中叶子结点的数目为 13，则度为 2 的结点数目为（ ）。

- A. 4
- B. 2
- C. 3
- D. 5

7. 设有无向图 $G=(V, E)$ 和 $G'=(V', E')$ ，如果 G' 是 G 的生成树，则下面不正确的说法是（ ）。

- I. G' 为 G 的连通分量
- II. G' 是 G 的无环子图
- III. G' 为 G 的极小连通子图，且 $V'=V$

- A. I、II
- B. II、III
- C. 只有 III
- D. 只有 I

8. 假定一组元素序列为 {38, 42, 55, 15, 23, 44, 34, 74, 45, 26}，按次序插入每个元素生成一棵平衡二叉树，那么最后得到的平衡二叉树中度为 2 的结点个数为（ ）。

- A. 1
- B. 3
- C. 4
- D. 5

9. 对以下关键字序列用快速排序进行排序，速度最慢的是（ ）。

- A. {19, 23, 3, 15, 7, 21, 28}
- B. {23, 21, 28, 15, 19, 3, 7}
- C. {19, 7, 15, 28, 23, 21, 3}
- D. {3, 7, 15, 19, 21, 23, 28}

10. 下列说法中正确的是（ ）。

- I. 对有 2500 个记录的索引顺序表（分块表）进行查找，最理想的块长为 50
- II. 顺序查找法只适合于顺序存储结构，不适合于链式存储结构
- III. 折半查找过程所对应判定树是一棵完全二叉树
- IV. 理想情况下，散列表的平均比较次数可达到 1 次

- A. I、IV
- B. II、III、IV
- C. III、IV
- D. I、II、III、IV

二、综合题

1. 设有两个用来存储大整数的链表 p 和 q 按照输入各位数字的逆序存放。设计一个算法，实现两个大整数的加法。

2. 设二叉树 T 采用二叉链表存储方式，根用 t 指针指示，设计一个算法，求指针 p 所指结点的双亲结点。

答案

一、选择题答案

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

二、综合题答案

1.

```
void addLargeInt(LNode* p, LNode* q, LNode*& result){

    LNode* tempP = p->next;
    LNode* tempQ = q->next;
    LNode* r = NULL;
    result = (LNode*)malloc(sizeof(LNode));
    result->next = NULL;
    r = result;
    int carry = 0;

    while (tempP!=NULL && tempQ != NULL)
    {
        LNode* temp = (LNode*)malloc(sizeof(LNode));
        temp->next = NULL;
        temp->data = (tempP->data + tempQ->data + carry)%10;
        carry = (tempP->data + tempQ->data + carry)/10;
        r->next = temp;
        r = r->next;
        tempP = tempP->next;
        tempQ = tempQ->next;
    }

    if (tempP == NULL) tempP = tempQ;

    while (tempP != NULL)
    {
        LNode* temp = (LNode*)malloc(sizeof(LNode));
        temp->next = NULL;
        temp->data = (tempP->data + carry)%10;
        carry = (tempP->data + carry)/10;
        r->next = temp;
        r = r->next;
        tempP = tempP->next;
    }

    if (carry > 0)
    {
        LNode* temp = (LNode*)malloc(sizeof(LNode));
        temp->next = NULL;
        temp->data = carry;
        r->next = temp;
    }
}
```

2.

```
BTNode* getParent(BTNode* t, BTNode* p)
```

```
{//只考虑 p 是正确的输入的情况
```

```
    if (t == NULL)
```

```
        return NULL;
```

```
    if (t == p)
```

```
        return NULL;
```

```
    if (t->lChild == p || t->rChild == p)
```

```
        return t;
```

```
    BTNode* parent = getParent(t->lChild, p);
```

```
    if (parent != NULL)
```

```
        return parent;
```

```
    else
```

```
        return getParent(t->rChild, p);
```

```
}
```

全套模拟卷以及答案解析视频讲解来辉解读公众号获取:

