

## 河海大学 2016 年硕士研究生入学考试初试试题 (A 卷)

科目代码: 838 科目名称: 数据结构及程序设计 满分: 150 分

注意: ①认真阅读答题纸上的注意事项; ②所有答案必须写在答题纸上, 写在试题纸或草稿纸上均无效; ③本试题纸须随答题纸一起装入试题袋中交回!

一、选择题: (2 分 $\times$ 15 = 30 分)

1、下列数据结构中, \_\_\_\_\_ 是非线性数据结构。

- A. 栈                      B. 队列                      C. 完全二叉树                      D. 字符串

2、程序段

```
for (i=n-1; i<1; i--)  
    for (j=1; j<I; j++)  
        if A[j] > A[j+1]  
            swap(A[j], A[j+1]);
```

其中  $n$  为正整数, 则最后一行语句执行, 在最坏情况下是\_\_\_\_\_。

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

3、设一个链表最常用操作是在末尾插入结点和删除尾结点, 则选用\_\_\_\_\_最节省时间。

- A. 带头结点的双循环链表                      B. 单循环链表  
C. 带尾指针的单循环链表                      D. 单链表

4、判定一个循环队列  $qu$  (最多元素为  $MaxSize$ ) 为空的条件是\_\_\_\_\_。

- A.  $qu \rightarrow rear - qu \rightarrow front == MaxSize$                       B.  $qu \rightarrow rear - qu \rightarrow front - 1 == MaxSize$   
C.  $qu \rightarrow rear == qu \rightarrow front$                       D.  $qu \rightarrow rear = qu \rightarrow front - 1$

5、若栈采用顺序存储方式存储, 现两栈共享空间  $V[1..m]$ ,  $top[i]$  代表第  $i$  个栈 ( $i=1, 2$ ) 栈顶, 栈 1 的底在  $v[1]$ , 栈 2 的底在  $V[m]$ , 则栈满的条件是\_\_\_\_\_。

- A.  $top[2] - top[1] = 0$                       B.  $top[1] + 1 = top[2]$   
C.  $top[1] + top[2] = m$                       D.  $top[1] = top[2]$

6、设  $A$  是  $n \times n$  的对称矩阵, 将  $A$  的对角线及对角线上方的元素以列为主的次序存放在一维数组  $B[1..n(n+1)/2]$  中, 对上述任一元素  $a_{ij}$  ( $1 \leq i, j \leq n$ , 且  $i \leq j$ ) 在  $B$  中的位置为\_\_\_\_\_。

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

7、已知一算术表达式中的中缀表达式为  $a - (b + c/d) * e$ , 其后缀形式为\_\_\_\_\_。

- A.  $-a + b * c/d$                       B.  $-a + b * cd/e$                       C.  $abcd/+e*-$                       D.  $-+*abc/de$

8、有  $n$  ( $n > 0$ ) 个分支结点的满二叉树的深度是\_\_\_\_\_。

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

9、下述二叉树中, 哪一种满足性质: 从任一结点出发到根的路径上所经过的结点序列按其关键字有序\_\_\_\_\_。

- A. 二叉排序树                      B. 哈夫曼树                      C. AVL 树                      D. 堆

- 10、图  $G$  是  $n$  个顶点的无向完全图，则下列说法错误的有\_\_\_\_\_。
- A.  $G$  的邻接多重表需要  $n(n-1)$  个边节点和  $n$  个顶点节点；
  - B.  $G$  的连通分量个数最少
  - C.  $G$  为连通图
  - D.  $G$  所用顶点的度的总和为  $n(n-1)$
- 11、下面说法错误的是\_\_\_\_\_。
- I. 求从指定源点到其余各顶点的 Dijkstra 最短路径算法中弧上权不能为负的原因是在实际应用中无意义；
  - II. 利用 Dijkstra 求每一对不同结点之间的最短路径的算法时间是  $O(n^3)$ ；（图用邻接矩阵表示）
  - III. Floyd 求每对不同结点对的算法中允许弧上的权为负，但不能有权和为负的回路。
- A. I、II、III      B. I      C. I、III      D. II、III
- 12、已知有向图  $G=(V, E)$ ，其中  $V=\{V_1, V_2, V_3, V_4, V_5, V_6, V_7\}$ ， $E=\{<V_1, V_2>, <V_1, V_3>, <V_1, V_4>, <V_2, V_5>, <V_3, V_5>, <V_3, V_6>, <V_4, V_6>, <V_5, V_7>, <V_6, V_7>\}$ ， $G$  的拓扑序列是\_\_\_\_\_。
- A.  $V_1, V_3, V_4, V_6, V_2, V_5, V_7$
  - B.  $V_1, V_3, V_2, V_6, V_4, V_5, V_7$
  - C.  $V_1, V_3, V_4, V_5, V_2, V_6, V_7$
  - D.  $V_1, V_2, V_5, V_3, V_4, V_6, V_7$
- 13、采用开放定址法解决冲突的散列查找中，发生聚集的主要原因是\_\_\_\_\_。
- A. 数据元素过多
  - B. 负载因子过大
  - C. 散列函数选择不当
  - D. 解决冲突的方法选择不当
- 14、下列排序算法中，在待排序数据已有序时，花费时间反而最多的是\_\_\_\_\_排序。
- A. 冒泡排序
  - B. 希尔排序
  - C. 快速排序
  - D. 堆排序
- 15、设线性表中每个元素有两个数据项  $k_1$  和  $k_2$ ，现对线性表按以下规则进行排序：先看数据项  $k_1$ ， $k_1$  值小的元素在前，大的在后；在  $k_1$  值相同的情况下，再看  $k_2$ ， $k_2$  值小的在前，大的在后。满足这种要求的排序方法是\_\_\_\_\_。
- A. 先按  $k_1$  进行直接插入排序，再按  $k_2$  进行简单选择排序
  - B. 先按  $k_2$  进行直接插入排序，再按  $k_1$  进行简单选择排序
  - C. 先按  $k_1$  进行简单选择排序，再按  $k_2$  进行直接插入排序
  - D. 先按  $k_2$  进行简单选择排序，再按  $k_1$  进行直接插入排序。
- 二、填空题（3 分  $\times$  12 + 2 分  $\times$  6 = 48 分，前 12 题，每空 3 分，后 6 题，每空 2 分）

- 1、线性表  $L=(a_1, a_2, \dots, a_n)$  用数组表示。假定删除表中任一元素的概率相同，则删除一个元素平均需要移动元素的个数是\_\_\_\_\_（1）\_\_\_\_\_。
- 2、对于一个具有  $n$  个结点的二叉树，当它为一棵\_\_\_\_\_（2）\_\_\_\_\_时，具有最小高度；当它为一棵\_\_\_\_\_（3）\_\_\_\_\_时，具有最大高度。
- 3、二叉树的后序遍历序列是 dabec，中序遍历序列是 debac，其前序遍历序列是\_\_\_\_\_（4）\_\_\_\_\_。
- 4、图 1 中给出了由 7 个结点组成的无向图。从结点 1 出发，对其进行深度优先遍历，



得到的序列是\_\_\_\_ (5) \_\_\_\_, 而进行广度优先遍历, 得到的顶点序列是\_\_ (6) \_\_。

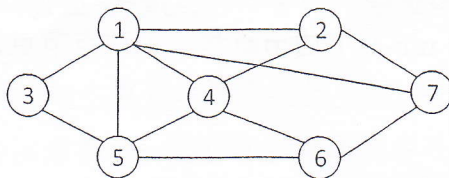


图 1: 第二题第 (4) 题

5、Prim 算法适用于求\_\_ (7) \_\_\_\_的图最小生成树; Kruskal 算法适用于求\_\_ (8) \_\_\_\_的图最小生成树。已知一个无向图, 如图 2 所示, 用 Prim 算法生成最小生成树 (设以结点 1 为起点) 是\_\_\_\_ (9) \_\_\_\_。

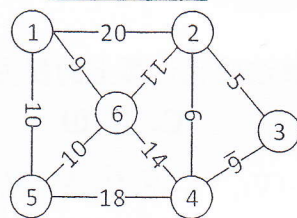


图 2: 第二题第 (5) 题

6、有数据  $WG=\{7, 19, 2, 6, 32, 3, 21, 10\}$ , 则所建哈夫曼树的树高是\_\_ (10) \_\_, 带权路径长度 WPL 为\_\_ (11) \_\_。

7、有向图  $G$  可拓扑排序的判别条件是\_\_ (12) \_\_。

8、以下程序的功能是实现带附加头结点的单链表数据结点逆序连接。请填空完善之。

```
void reverse (pointer h){
    pointer p, q;
    p = h->next; h->next = NULL;
    while (____ (13) ____){
        q = p;
        p = p->next;
        (____ (14) ____ )
        h->next = q;
    }
}
```

9、下面程序功能是将二叉树  $bt$  中的每一结点的左右子树互换, 其中  $ADDQ(Q, bt)$ 、 $DELQ(Q)$ 、 $EMPTY(Q)$  分别为进队、出队和判别队列是否为空的函数。请填写下面算法中空白之处。

```
typedef struct node{
    int data;
    struct node *lnode, *rnode;
}btnode;
void EXCHANGE(btnode *bt){
```

```

btnode *p, *q;
if (bt){
    ADDQ(Q, bt);
    while ( (15) ){
        p=DELQ(Q);
        if(p→lchild) (16);
        if(p→rchild) ADDQ(Q, p→rchild);
        q = (17);
        p→rchild = (18);
        p→lchild = q;
    }
}
}

```

计算机/软件工程专业  
每个学校的  
考研真题/复试资料/考研经验  
考研资讯/报录比/分数线  
免费分享



微信 扫一扫  
关注微信公众号  
计算机与软件考研

### 三、应用题 (9 分+9 分+12 分=30 分)

- 1、如果需要一个具有  $n$  个元素的集合中找到第  $k$  ( $1 \leq k \leq n$ ) 个最小元素, 根据你所学习掌握的排序方法, 哪种方法最合适? 给出实现的思想。
- 2、已知序列 {17, 31, 13, 11, 20, 35, 25, 8, 4, 24, 40, 27}, 请画出该序列的二叉排序树, 并分别给出下列操作后的二叉树;  
(1) 插入数据9; (2) 再删除结点17。
- 3、已知一组关键字为 {26, 36, 41, 38, 44, 15, 68, 12, 6, 51, 25}, 用链地址法解决冲突, 假设装填因子  $\alpha = 0.75$ , Hash 函数的形式为  $H(\text{key}) = \text{key} \% P$ , 回答以下问题:  
(1) 构造出 Hash 函数;  
(2) 分别计算等概率情况下查找成功和查找失败的平均查找长度。

### 四、算法设计题 (12 分+14 分+16 分=42 分)

- 1、设有一个单链表  $L$  拆成两个单链表, 其中  $L_1$  为头的链表保持原链表  $L$  向后的链接, 另一个链表的头为  $L_2$ , 其链接方向与原链表  $L$  相反。  $L_1$  包含原链表的奇数序号的节点,  $L_2$  包含原链表的偶数序号的节点。请写出实现该链表拆分的算法思想, 以及相应实现的算法。
- 2、已知有一棵用二叉链表示的二叉树, 编写程序, 输出从根节点到叶节点的最长路径上的所有节点, 并写出算法思想。
- 3、编写算法求图的中心点。设  $V$  是有向图  $G$  的一个结点,  $v$  的偏心度定义为:  $\max\{\text{从 } w \text{ 到 } v \text{ 的最短路径} \mid w \text{ 是图 } G \text{ 中的所有结点}\}$ , 如果  $v$  是有向图  $G$  中具有最小偏心度的结点, 则称结点  $v$  是  $G$  的中心点。