

**杭州电子科技大学**  
**2014 年攻读硕士学位研究生入学考试**  
**《数据结构》试题**

(试题共 六 大题, 共 6 页, 总分 150 分)

姓名\_\_\_\_\_报考专业\_\_\_\_\_准考证号\_\_\_\_\_

**【所有答案必须写在答题纸上, 做在试卷或草稿纸上无效】**

**一、是非题 (T 正确, F 错误, 本大题共 5 小题, 每小题 2 分, 本大题共 10 分)**

1. 邻接表可以用以表示无向图, 也可用以表示有向图。( )
2. 算法的优劣与算法描述语言无关, 但与所用计算机有关。( )
3. 二叉树是一棵结点的度最大为 2 的树。( )
4. 队列是与线性表完全不同的一种数据结构。( )
5. 对于任何待排序序列来说, 快速排序均快于起泡排序。( )

**二、选择题 (本大题共 14 小题, 每小题 2 分, 本大题共 28 分)**

1. 递归程序可借助于( )转化为非递归程序。  
A. 线性表      B. 栈      C. 队列      D. 数组
2. 对二叉排序树按( )可得到有序序列。  
A. 层次遍历      B. 先序遍历      C. 中序遍历      D. 后序遍历
3. 顺序存储设计时, 存储单元的地址( )。  
A. 一定连续      B. 一定不连续      C. 不一定连续      D. 部分连续, 部分不连续
4. 已知一算术表达式的中缀形式为  $A+B*C-D/E$ , 后缀形式为  $ABC*+DE/-$ , 其前缀形式为( )。  
A.  $-A+B*C/DE$       B.  $-A+B*CD/E$       C.  $-+*ABC/DE$       D.  $-+A*BC/DE$
5. 在有向图 G 的拓扑序列中, 若顶点  $V_i$  在顶点  $V_j$  之前, 则下列情形不可能出现的是( )。  
A. G 中有弧  $\langle V_i, V_j \rangle$       B. G 中没有弧  $\langle V_j, V_i \rangle$   
C. G 中没有弧  $\langle V_i, V_j \rangle$       D. G 中有弧  $\langle V_j, V_i \rangle$
6. 下列说法不正确的是( )。  
A. 图的遍历是从图中某一顶点出发访遍图中其余顶点, 且使每一个顶点仅被访问一次。  
B. 图的遍历的基本算法有两种: 深度优先搜索和广度优先搜索。  
C. 图的深度优先搜索不适用于有向图。  
D. 图的深度优先搜索是一个递归过程。

7.  $n$  个结点的有向完全图含有弧的数目 ( )。
- A.  $n*n$       B.  $n(n+1)$       C.  $n/2$       D.  $n*(n-1)$
8. 在下面的程序段中，“ $x=x+1$ ”语句的频度为 ( )。
- ```
for(int i=0;i<n;i++)
    for(int j=0;j<n;j++)
        x=x+1;
```
- A.  $O(2n)$       B.  $O(n)$       C.  $O(n^2)$       D.  $O(\log_2 n)$
9. 静态链表中指针用 ( ) 代替，表示结点在数组中的相对位置。
- A. 内存地址      B. 数组下标      C. 下一元素地址      D. 左、右孩子地址
10. 某内排序方法的稳定性是指 ( )。
- A. 该排序算法不允许有相同的关键字记录
- B. 该排序算法允许有相同的关键字记录
- C. 平均时间为  $O(n \log n)$  的排序方法
- D. 假设关键字  $K_i = K_j$  ( $1 \leq i \leq n, 1 \leq j \leq n, i \neq j$ )，且在排序前的序列中  $R_i$  领先于  $R_j$  (即  $i < j$ )，在排序后的序列中， $R_i$  仍领先于  $R_j$
11. 一组记录的关键码为 (47, 80, 57, 39, 41, 85)，则利用快速排序的方法，以第一个记录为基准得到的一次划分结果为 ( )。
- A. (39, 41, 47, 57, 80, 85)      B. (41, 39, 47, 80, 57, 85)
- C. (41, 39, 47, 57, 80, 85)      D. (41, 39, 47, 85, 57, 80)
12. 比较次数与序列的原始状态无关的排序方法是 ( ) 排序法。
- A. 插入      B. 选择      C. 起泡      D. 快速
13. 适用于折半查找的表的存储方式及元素排列要求为 ( )
- A. 链接方式存储，元素无序      B. 链接方式存储，元素有序
- C. 顺序方式存储，元素无序      D. 顺序方式存储，元素有序
14. 下面关于 B 和 B+树的叙述中，不正确的是 ( )
- A. B 树和 B+树都是平衡的多叉树。
- B. B 树和 B+树都可用于文件的索引结构。
- C. B 树和 B+树都能有效地支持顺序检索。
- D. B 树和 B+树都能有效地支持随机检索。

### 三、填空题 (本大题共 11 空，每空 2 分，本大题共 22 分)

1. 森林的两类遍历方法为先序遍历与\_\_\_\_\_；先根遍历树可以借用二叉树的\_\_\_\_\_遍历算法实现；后根遍历树可以借用二叉树的\_\_\_\_\_遍历算法实现。
2. 已知一无向图  $G=(V, E)$ ，其中  $V=\{a, b, c, d, e\}$   $E=\{(a, b), (a, d), (a, c), (d, c), (b, e)\}$  现用某一种图遍历方法从顶点  $a$  开始遍历图，得到的序列为  $abecd$ ，则采用的是\_\_\_\_\_遍历方法。
3. 将整型数组  $A[7][7]$  按行优先次序存储在起始地址为 1000 的连续的内存单元中，设每个整型元素占 2 字节，则元素  $A[6][1]$  的地址是：\_\_\_\_\_。

4. 设一棵二叉树 BT 的存储结构如下：

|        | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|--------|---|---|---|---|---|---|---|---|
| lchild | 2 | 3 | 0 | 0 | 6 | 0 | 0 | 0 |
| data   | A | B | C | D | E | F | G | H |
| rchild | 0 | 5 | 4 | 0 | 8 | 7 | 0 | 0 |

其中 lchild, rchild 分别为结点的左、右孩子指针域, data 为结点的数据域。  
则该二叉树的高度为\_\_\_\_; 第 3 层有\_\_\_\_个结点 (根结点为第 1 层)。

5. 当广义表中的每个元素都是原子时, 广义表便成了\_\_\_\_\_。

6. 栈是限定在表尾进行插入或删除操作的线性表, 栈对数据元素的操作是按\_\_\_\_原则进行的。设有一个空栈, 现有输入序列为 1, 2, 3, 4, 5, 经过 PUSH, PUSH, POP, PUSH, POP, PUSH, PUSH 之后, 输出序列是\_\_\_\_\_。

7. 对于一个具有 n 个结点的单链表, 在表中插入 1 个结点的时间复杂度为\_\_\_\_\_。

#### 四、图示结构题 (本大题共 5 小题, 每小题 8 分, 本大题共 40 分)

1. 已知某森林的先序遍历次序为: A, D, E, F, G, H, B, I, C, J, K, L, M, N

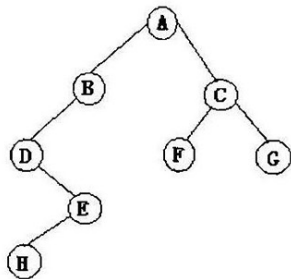
中序遍历次序为: D, F, G, H, E, A, I, B, J, L, M, N, K, C

1) 画出该森林

2) 画出该森林用孩子兄弟法表示的存储结构

2. 根据插入次序 (82, 92, 102, 112, 87, 72, 77, 63, 74) 建立二叉排序树。并根据该插入次序建立平衡二叉树。

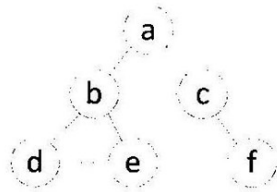
3. 对下图所示二叉树分别按前序、中序、后序遍历, 给出相应的结点序列, 并画出中序线索二叉树。



4. 采用哈希函数  $H(k) = 3 * k \bmod 13$  并用开放地址法处理冲突, 增量序列选取采用线性探测再散列方式, 在数地址空间  $[0..12]$  中对关键字序列 22, 41, 53, 46, 30, 13, 1, 67, 51

1) 构造哈希表 (画示意图); 2) 装填因子; 3) 查找成功时的平均查找长度; 4) 查找不成功时的平均查找长度。

5. 已知某无向图如下图所示：画出两类图的存储结构示意图 1) 数组表示法（邻接矩阵）存储结构，2) 邻接表存储结构。



五、阅读以下函数，指出算法的功能（本大题共 5 小题，每小题 6 分，本大题共 30 分）

1. Status A1(SqList L, ElemType cur\_e, ElemType &next\_e) // L 为循序线性表

```

{
    int i=1;
    ElemType *p=L.elem;
    while(i<L.length&&*p!=cur_e)
    {
        i++;
        p++;
    }
    if(i==L.length)
        return INFEASIBLE;
    else
    {
        next_e=**p;
        return OK;
    }
}

```

2. Status A2(SqStack &S, SElemType e) // S 为顺序栈

```

{
    if(S.top-S.base>=S.stacksize)
    {
        S.base=(SElemType*)realloc(S.base, (S.stacksize+STACKINCREMENT)*sizeof(SElemType));
        if(!S.base)
            exit(OVERFLOW);
        S.top=S.base+S.stacksize;
    }
}

```

```

        S.stacksize+=STACKINCREMENT;
    }
    *(S.top)++=e;
    return OK;
}

3. int A3(PTree T)//T 为双亲表存储的树
{
    int k,m,def,max=0;
    for(k=0;k<T.n;++k)
    {
        def=1;
        m=T.nodes[k].parent;
        while(m!=-1)
        {
            m=T.nodes[m].parent;
            def++;
        }
        if(max<def)
            max=def;
    }
    return max;
}

4. void A4(SSTable &ST)// ST 为静态顺序查找表
{
    int i,j,k;
    for(i=1;i<ST.length;i++)
    {
        k=i;
        ST.elem[0]=ST.elem[i];
        for(j=i+1;j<=ST.length;j++)
            if LT(ST.elem[j].key,ST.elem[0].key)
            {
                k=j;
                ST.elem[0]=ST.elem[j];
            }
        if(k!=i)
        {
            ST.elem[k]=ST.elem[i];

```

```

        ST.elem[i]=ST.elem[0];
    }
}

5. void A5(Sqlist &L)//L 为顺序表
{
    int i, j;
    for(i=2; i<=L.length; ++i)
        if (LT(L.r[i].key, L.r[i-1].key))
        {
            L.r[0]=L.r[i];
            for(j=i-1; LT(L.r[0].key, L.r[j].key); --j)
                L.r[j+1]=L.r[j];
            L.r[j+1]=L.r[0];
        }
}

```

#### 六、算法设计题（本大题共 2 小题，每小题 10 分，本大题共 20 分）

1. 实现顺序线性表基本操作-获取 1 个数据元素的前驱。

2. 实现图遍历算法中的深度优先搜索算法。