

浙江师范大学 2006 年硕士研究生 入学考试试题

考试科目：428 数据结构

报考学科、专业：课程与教学论（信息技术教育）、计算机软件与理论

一、判断题（下列各题你认为正确的，请在前面的括号内打√，错误的打×，每题 1 分，共 15 分）

1. 数据结构的概念包括数据的逻辑结构、数据在计算机中的存储方式和数据的运算三个方面。
2. 用顺序表来存储线性表时，不需要另外开辟空间来保存数据元素之间的相互关系。
3. 单链表的结点插入和删除的时间复杂度均为 $O(n^2)$ 。
4. 栈是一种先进先出的线性表。
5. 串是一个或多个字符组成的有限序列。
6. 完全二叉树的叶子结点只可能在层次最大的一层上出现。
7. 图可以没有边，但不能没有顶点。
8. 在有向图 G 中， $\langle V_2, V_1 \rangle$ 和 $\langle V_1, V_2 \rangle$ 是两条不同的边。
9. 快速排序算法是稳定的排序，而希尔排序是不稳定的。
10. 折半查找方法要求待查表必须是顺序存储结构的有序表。
11. 从循环单链表的任一结点出发，不一定能找到表中所有结点。
12. AOE-网中路径长度最长的路径叫做关键路径。
13. 赫夫曼树是带权路径长度最长的树。
14. 文件是大量性质相同的记录组成的集合。
15. 一个广义表可以是另一个广义表的子表。

二、填空题（每空 1 分，共 15 分）

1. 算法是对特定问题求解步骤的一种描述，是指令的有限序列。一个算法具有以下重要特征：具有零个或多个输入量、具有一个或多个输出量、（1）、（2）、和可行性。
2. 数据结构是相互之间存在一种或多种特定关系的数据元素的集合，根据元素之间关系的不同特性，通常有四类基本结构：集合、（3）、（4）和网状结构。
3. 树所对应的二叉树，其根结点的 （5）子树一定为空。
4. 将插入限定在表的一端，而删除限定在表的另一端进行的线性表称为 （6）；允许插入的一端称为 （7）；所有插入和删除都在表的一端进行的线性表称为 （8）。

5. n 个顶点的有向完全图具有 (9) 条弧。
6. n 个元素的顺序查找的平均查找长度为 (10)。
7. n 个结点的完全二叉树, 其深度 $h =$ (11)。
8. 由某个集合上的偏序得到该集合上的一个全序, 这个操作称之为 (12)。
9. 二叉树的叶结点数 n_0 与二度结点数 n_2 的关系是 (13)。
10. 二叉排序树若右子树非空, 则右子树上所有结点的值均 (14) 它的根结点的值; 若左子树非空, 则左子树上所有结点的值均 (15) 它的根结点的值。

三、单选题 (本题的每一备选答案中, 只有一个是正确的, 请选择你认为正确的答案, 每小题 3 分, 共 33 分)

1. 设单链表中指针 p 指着结点 A , 若要删除 A 之后的结点 (假设存在), 则需要修改指针的操作为_____。
 A. $p \rightarrow next = p \rightarrow next \rightarrow next$ B. $p = p \rightarrow next$
 C. $p = p \rightarrow next \rightarrow next$ D. $p \rightarrow next = p$
2. 如果一棵树的结点 A 有 3 个兄弟结点 (不含 A), B 为 A 的双亲结点, 则结点 B 的度为_____。
 A. 3 B. 4 C. 5 D. 1
3. 广义表 $A = (a, b, (c, d), (e, (f, g)))$, 则下面式子的值为_____。
 $GetHead (GetTail (GetHead (GetTail (GetTail (A)))))$
 A. (g) B. d C. e D. (d)
4. 若进栈序列为 1, 2, 3, 4, 则不可能得到的出栈序列是 _____。
 A. 3, 2, 1, 4 B. 3, 2, 4, 1 C. 4, 2, 3, 1 D. 2, 4, 3, 1
5. 对于下列二叉树 (图 1), 其中序序列为 _____。
 A. DBEAFCCG B. DEBFGCA C. GFCEBDA D. DBEACFG
6. 对于下列 AOV 网 (图 2), 不能出现的拓扑序列为 _____。
 A. 1 2 3 4 5 B. 1 2 4 3 5 C. 2 4 1 3 5 D. 2 1 4 3 5

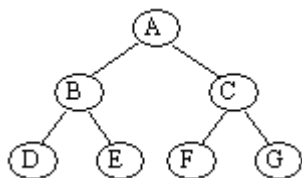


图 1

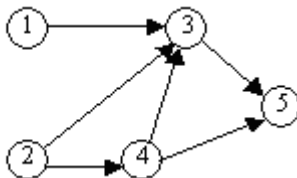


图 2

7. 设二叉树的根为第一层, 则第 i 层上的结点数最多有 _____。
 A. 2^i B. 2^{i+1} C. 2^{i-1} D. 2^{i-1}
8. 衡量查找算法效率的主要标准是 _____。
 A. 元素个数 B. 所需的存贮量
 C. 平均查找长度 D. 算法难易程度

9. 给定下列有向图（图3），从顶点1出发，其深度优先搜索序列为_____

A. 12534

B. 12435

C. 14325

D. 12345

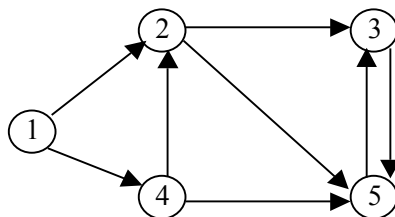


图 3

10. 简单选择排序的时间复杂度为_____

A. $O(n)$

B. $O(n \log_2 n)$

C. $O(n^2)$

D. $O(\log_2 n)$

11. 与线性表的链接存贮不相符合的特性是_____

A. 便于插、删运算

B. 存贮空间动态分配

C. 需要连续的存贮空间

D. 只能顺序查找

四、简答和过程描述题（共 50 分）

1.（7 分）分别画出具有 3 个结点的树和 3 个结点的二叉树的所有不同形态。

2.（9 分）考虑图 5 所示的带权无向图：

（1）从顶点 A 出发，求它的深度优先生成树；（2 分）

（2）从顶点 E 出发，求它的广度优先生成树；（2 分）

（3）根据普里姆（prime）算法求它的最小生成树，并画出构造过程。（5 分）

3.（4 分）已知一棵二叉树的先根次序访问序列为 EBADCFHGIKJ，中序访问序列为 ABCDEFGHIJK，请画出该二叉树。

4.（10 分）对于下图（图 6）所示的带权图，利用 Dijkstra 算法求从源点 V1 到其余各顶点的最

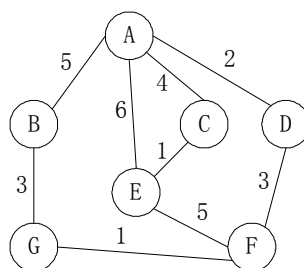


图 5

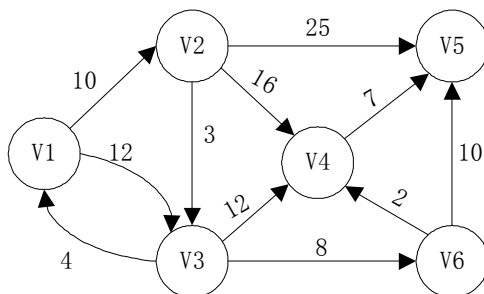


图 6

短路径及其长度，并写出算法求解过程（画表表示）。

5. (8分) 在地址空间为 0-16 的散列区中, 对以下关键字序列用不同的冲突处理方法分别构造哈希表: (Jan, Feb, Mar, Apr, May, June, July, Aug, Sep, Oct, Nov, Dec)

(1) 用线性探测开放定址法处理冲突;

(2) 用链地址法处理冲突;

假设哈希函数 $H(X) = \lfloor i/2 \rfloor$, 其中 $\lfloor \quad \rfloor$ 为取整运算, i 为关键字中第一个字母在字母表中的序号。

6. (6分) 对下列关键字序列进行快速排序, 请写出每一趟排序的结果, 并标识出在第一趟排序过程中数据交换的情况。

35 92 15 19 10 80 100, 30

7. (6分) 序列 {12, 36, 24, 85, 47, 30, 53, 92} 所对应的完全二叉树为堆, 画图说明第一次输出堆顶元素后调整建新堆的过程。(只需第一次即可)

五、算法设计题 (37 分)

1. (8分) 设顺序表 VA 中的数据元素递增有序, 试设计算法并编写代码, 将 x 插入到顺序表的适当位置上, 以保持该表的有序性; 假设结点数据类型为 ElemType, 顺序表结构为

```
typedef struct List{
    int listsize; //listsize 为顺序表最大的长度
    ElemType elem[listsize];
    int length; //顺序表的实际长度
}
```

函数原型为 InsertOrderList(struct List a, ElemType x)

2. (10分) 假设链表结点数据类型为 ElemType, 线性链表结构定义为

```
typedef struct LNode{
    ElemType data;
    struct LNode *next;
};
```

试设计算法并编写代码实现以下功能:

- (1) (5分) 在带头结点的单链表 L 上实现查找值为 x 的结点, 并返回指向该结点的指针, 若找不到返回 NIL; 函数原型为

```
struct LNode *LOCATE(struct LNode *L, x);
```

- (2) (5分) 在带头结点的单链表 L 上实现删除值为 x 的所有结点, 返回删除结点的个数; 函数原型为 int List_Delete(struct LNode *L, x);

3. (9分) 已知二叉排序树中任意分支结点的值均大于其左孩子结点的值 (若左孩子结点存在), 并且小于或等于其右孩子结点的值 (若右孩子结点存在)。设二

又排序树采用二叉链表存储结构，链结点的构造为：

```
typedef struct BiTNode{
    ElemType data;
    struct BiTNode *lchild, *rchild; //左右子树
};
```

根结点的地址由 T 指出。下面是在该二叉排序树中查找值为 key 的非递归算法，查找成功，返回被查到结点的地址，否则返回 NIL，请在算法的空缺处填入适当的内容。

```
struct BiTNode *SearchBST(struct BiTNode *T, ElemType key) {
    p=T;
    while( ! p )
    {
        .....
        请补充完整
        .....
    }
    return NIL;
} // SearchBST
```

4. (10 分) 编写算法，求得所有包含在串 S 中而不包含在串 T 中的字符 (S 中重复的字符只选一个) 所构成的新串 R, 以及 R 中每个字符在 S 中第一次出现的位置。