



科目代码: 896

科目名称: 程序设计与数据结构

```

do {
    while (p!=NULL) {
        if ( )
            STACK_FULL("Stack is overfull");
        top=top+1; //进栈//
        ( );
        p = p->LeftChild;
    };
    p=STACK[top]; //退栈//
    ( );
    VISIT(p); //访问一个结点//
    p = p->RightChild;
while ( );
}

```

## 二、判断下面各陈述是否正确, 并说明理由 (每题 4 分, 共 24 分)

- 1、使用三元组表表示稀疏矩阵可以节省存储空间。
- 2、度为 2 的树是二叉树。
- 3、当在函数定义中作为形参时, `char s[]` 和 `char *s` 是等价的。
- 4、二叉树叶结点的数目只与度为 2 的结点的数目有关。
- 5、对于有序链表可以采用折半查找。
- 6、在执行某种排序算法的过程中, 出现了数据元素朝着与其在最终排序结果中的位置相反方向移动的情况, 因此断定该排序算法是不稳定的。

## 三、简答题 (每题 5 分, 共 30 分)

- 1、已知有实现同一功能的两个算法, 其时间复杂度分别为  $O(2^n)$  和  $O(n^{10})$ , 假设现实计算机可连续运行的时间为  $10^7$  秒 (约 100 天), 又每秒可执行基本操作为  $10^5$  次。试问在此条件下, 这两个算法可解决问题的规模 (即  $n$  的范围) 各为多少? 哪个算法更合适? 试说明理由。
- 2、已知广义表  $L = ((a, b), (c, (d, (e))), f)$ 
  - (1) 试利用广义表取表头  $head(L)$  和取表尾  $tail(L)$  的基本运算, 将原子  $c$  从广义表  $L$  中分解出来, 请写出运算表达式。

科目代码: 896

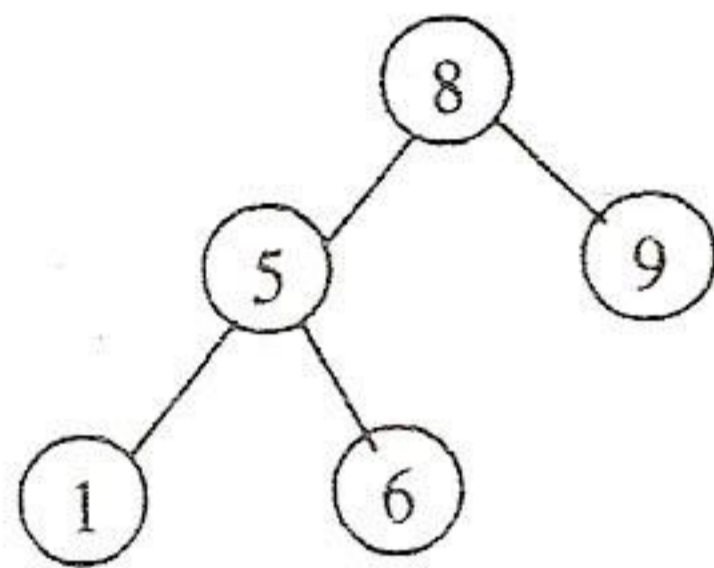
科目名称: 程序设计与数据结构

(2) 请给出下列表达式的运算结果:

head(tail(head(tail(L))))

tail(tail(head(L)))

- 3、能够生成下图所示的二叉排序树的关键字初始序列有几种? 写出所有这些序列组合。



- 4、已知一带权连通图采用邻接矩阵存储方法, 并且邻接矩阵采用三元组表来表示, 其中第一个三元组(5,5,16)分别表示邻接矩阵的行数、列数与非零元素的个数, 从第二个三元组开始, 依次按行序为主序的次序分别给出 16 个非零元素, 它们依次为(1,2,7), (1,3,6), (1,4,9), (2,1,7), (2,3,8), (2,4,4), (2,5,4), (3,1,6), (3,2,8), (3,4,6), (4,1,9), (4,2,4), (4,3,6), (4,5,2), (5,2,4), (5,4,2)。请分别画出该带权连通图的两棵最小生成树。
- 5、将数组 13,5,10,7,27,9,4,15,33,20 调整成极小堆, 画出这个极小堆的逻辑图和内存映像。
- 6、设某有序连续顺序文件的记录按关键字值从小到大排列, 请用文字叙述在该文件中采用折半查找方法确定一个记录存在与否的过程。

#### 四、改错 (每题 5 分, 共 10 分)

- 1、假设带头节点的单链表中只设一个指针指向头节点, 下面算法将实现以下功能: 在单链表中查找值为 value 的节点并删除。

```

typedef int DataType;
typedef struct node {
    DataType data;
    struct node *link;
}ListNode;
ListNode *head;
  
```

科目代码: 896

科目名称: 程序设计与数据结构

```

void DeleteNode(DataType value){
    ListNode *q, *p=head;
    while (p!=NULL) {
        if(value==p->link->data){
            q=p->link;
            delete q;
            p->link = q->link;
        }
        p=p->link;
    }
}

```

- 2、已知二叉树的存储结构为二叉链表, 下面算法将实现以下功能: 将二叉树 T 中分支节点按照后续遍历序列逆序保存在单链表 L 中。

```

typedef int DataType;
typedef struct listnode {
    DataType data;
    struct listnode *link;
}ListNode;
typedef ListNode* LinkList;
typedef struct treenode {
    DataType data;
    struct treenode *LeftChild, *RightChild;
} TreeNode;
typedef TreeNode* BiTree;

LinkList L;
void PostorderList(BiTree T) {
    LinkNode *p;
    if (T){
        PostorderList (T->LeftChild);
        PostorderList (T->RightChild);
        if( (! T->LeftChild) && (! T->RightChild)){
            p->Data=T->Data;
            p->Link=L;
            L=p;
        }
    }
}

```

科目代码: 896

科目名称: 程序设计与数据结构

## 五、应用题 (共 48 分。若需编程, 可任选自己熟悉的语言实现)

- 1、 (12 分) 已知一个非空的线性链表的首结点地址由 list 指出, 请编写程序, 将链表中数据域值最小的那个结点移到链表的最前面去。
- 2、 (16 分) 已知某二叉树有  $n$  个结点, 各结点存放的值是互不相同的字符, 其先序遍历和中序遍历的序列分别存放在数组 pre 和 in 中。
  - (1) 编写一个函数建立该树的二叉链表。
  - (2) 假设已知二叉树的前序遍历序列为 ABCFEHIDG, 中序遍历序列为 ECHFIBDGA, 请画出该二叉树, 并给出它的后序遍历序列。
- 3、 (20 分) 有一只小袋鼠路过一条河, 看到河中一块石头上有一个布娃娃, 于是它想跳过去把布娃娃拣起来玩。可是那块石头离它的距离超出了它所能跳的最远距离, 因此小袋鼠决定把河中其他一些石头当作中继站, 这样它就可以每次跳比较短的距离 (但需要跳多次), 最后到达布娃娃所在的那块石头上。在这样一串石头间连续跳跃, 显然小袋鼠一次能够跳跃的距离必须至少等于这串石头间间隔最远的距离, 这一距离称为该串石头间的跳跃距离。例如, 如果在袋鼠跳跃路径上各石头之间的间距分别为 2, 1, 3, 3, 5 和 2, 则这里的跳跃距离为 5。

输入: 第一个输入数据是一个整数  $n$  ( $2 \leq n \leq 200$ )。接下来的  $n$  组数据是  $n$  组二维坐标。其中第一组为小袋鼠所在位置的坐标, 第二组为布娃娃所在石头的坐标, 其余  $n-2$  组为可以当作中继站的其他石头的坐标。

输出: 袋鼠到达布娃娃所要跳的最小跳跃距离。

  - (1) 按照上述的输入、输出要求编写解决该问题的算法。
  - (2) 用下面的输入实例给出算法的运行过程及结果。

输入实例:

5, (2, 4), (4, 5), (3, 5), (1, 2), (4, 1)