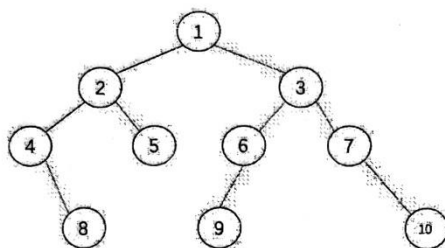


## 2018-2019《数据结构》期中考试

一、一棵二叉树如右图所示

1. 写出该树的先序遍历序列 (6 分)。
2. 写出该树的中序遍历序列 (6 分)。
3. 写出该树的后序遍历序列 (6 分)。



二、设单链表的结点结构为 (data, next), next 为指针域, 在单链表

中指针为 P 的结点之后插入指针为 S 的结点

- (a) 用 C 语言写出操作步骤; (8 分)
- (b) 以上操作复杂度如何? 写出大 O 表示的复杂度。 (3 分)
- (c) 如改用顺序表储存, 在某随机位置插入结点, 复杂度如何? (3 分)

小麦铺超市内复印店出售

三.已知二叉树用二叉链表存储, 写一个非递归算法, 将所有结点的  
左右子树互换 (建议先序遍历, 左右子树交换)

(32 分)

```
void ExchangeChild (BTreeNode <Type> *T )
{
    SeqStack <BTreeNode <Type> *> S; /*顺序栈 存放指针*/
    BTreeNode <Type> *p = T; //交换左右子树时的临时指针
    while (p || ! S.IsEmpty ( )) //若堆栈非空, 则必有节点右子树未访问
    { ... ..
```

小麦铺超市内复印店出售

四、假设有两个按元素值递增次序排列的带头结点的线性表 (la 和 lb),

均以单链表形式存储。请编写算法将这两个单链表归并为一个按元素值递减次

序排列的单链表 (la 作结果链表的头指针),

并要求利用原来两个单链表的结点存放归并后的单链表

(36 分)

提示 / 备注:

链表结点类型是 struct LinkListNode 包含

数据 + 指向自己类型的指针 next

函数应当有两个参数 LinkListNode la, lb;

原来两个单链表的头结点 (单独头结点)

```
LinkListNode Union (LinkListNode * la, lb)
{
    _____//数据声明
    pa=la->next;pb=lb->next;//pa, pb 分别是链表 la 和 lb
    的工作指针 la->next=null;
    //la 作结果链表的头指针, 先将结果链表初始化为空
    while (pa != null && pb != null) //可以有适当注释
    {
        if ( _____ <= _____ )
        {
            _____} //多个语句
        }
        else
        {
            _____} //多个语句
    }
    while (pa != null)
```

小麦铺超市内复印店出售

```
{ _____ } //多个语句  
while (pb! =null)
```

```
{ _____ } //多个语句  
return la;  
}  
//算法 Union 结束。
```

小麦铺超市内复印店出售