二叉树三种遍历的非递归算法（背诵版）

本贴给出二叉树先序、中序、后序三种遍历的非递归算法，此三个算法可视为标准算法，直接用于考研答题。

1.先序遍历非递归算法

```
#define maxsize 100
typedef struct
{
    Bitree Elem[maxsize];
    int top;
}SqStack;

void PreOrderUnrec(Bitree t)
{
    SqStack s;
    StackInit(s);
    p=t;

    while (p!=null || !StackEmpty(s))
    {
       while (p!=null)              //遍历左子树
       {
           visite(p->data);
           push(s,p);
           p=p->lchild;
       }//endwhile

       if (!StackEmpty(s))          //通过下一次循环中的内嵌 while 实现右子树遍历
       {
           p=pop(s);
           p=p->rchild;
       }//endif

    }//endwhile

}//PreOrderUnrec
```

2.中序遍历非递归算法

```
#define maxsize 100
typedef struct
{
    Bitree Elem[maxsize];
```

```
    int top;
}SqStack;

void InOrderUnrec(Bitree t)
{
    SqStack s;
    StackInit(s);
    p=t;
    while (p!=null || !StackEmpty(s))
    {
        while (p!=null)              //遍历左子树
        {
            push(s,p);
            p=p->lchild;
        }//endwhile

        if (!StackEmpty(s))
        {
            p=pop(s);
            visite(p->data);        //访问根结点
            p=p->rchild;             //通过下一次循环实现右子树遍历
        }//endif

    }//endwhile

}//InOrderUnrec
```

3.后序遍历非递归算法

```
#define maxsize 100
typedef enum{L,R} tagtype;
typedef struct
{
    Bitree ptr;
    tagtype tag;
}stacknode;

typedef struct
{
    stacknode Elem[maxsize];
    int top;
}SqStack;

void PostOrderUnrec(Bitree t)
```

```
{
    SqStack s;
    stacknode x;
    StackInit(s);
    p=t;

    do
    {
       while (p!=null)        //遍历左子树
       {
           x.ptr = p;
           x.tag = L;         //标记为左子树
           push(s,x);
           p=p->lchild;
       }

       while (!StackEmpty(s) && s.Elem[s.top].tag==R)
       {
           x = pop(s);
           p = x.ptr;
           visite(p->data);   //tag 为 R，表示右子树访问完毕，故访问根结点
       }

       if (!StackEmpty(s))
       {
           s.Elem[s.top].tag =R;     //遍历右子树
           p=s.Elem[s.top].ptr->rchild;
       }
    }while (!StackEmpty(s));
}//PostOrderUnrec
```