

# 《数据结构》上机报告

2018 年 11 月 6 日

姓名：张天然 学号：1751237 班级：计 2 得分：

试验题目	二叉树
实验目的	理解二叉树的逻辑结构和存储结构，熟练掌握二叉树的相关操作。
问题描述	二叉树是 $n$ ( $n \geq 0$ ) 个结点的有限集合，它或为空树，或是由一个称之为根的结点加上两棵分别称为左子树和右子树的互不相交的二叉树组成。对二叉树的操作主要有：建立一棵二叉树；遍历二叉树；求二叉树的深度、结点数等。
基本要求	1、定义二叉链表类型，实现二叉树的基本操作：建立、先中后序遍历、层次遍历及输出。 2、求二叉树的深度、总结点数、叶子结点数、二叉树的复制等。 3、用非递归完成二叉树的中序遍历。
选作要求	无
数据结构设计	<pre>struct BiTNode {     T data;     BiTNode *lchild, *rchild;} private:     BiTNode&lt;T&gt; *root;</pre>
功能(函数)说明	<pre>BiTree();    //构造函数 bool IsEmpty();    //判断是否为空 void CreateBiTree(T space, T fin);    //建立二叉树 void ClearBiTree(BiTNode&lt;T&gt;* root);    //清空二叉树 void display1();    //先序输出 void display2();    //中序输出 void display3();    //后序输出 void PreOrderTraverse(BiTNode&lt;T&gt; *temp);    //先序遍历 void InOrderTraverse(BiTNode&lt;T&gt; *temp);    //中序遍历 void PostOrderTraverse(BiTNode&lt;T&gt; *temp);    //后序遍历 void LevelOrderTraversal(BiTNode&lt;T&gt;* root);    //层序遍历 int Count(BiTNode&lt;T&gt;* root);    //求结点数 int Depth(BiTNode&lt;T&gt;* root);    //求深度 int Leave(BiTNode&lt;T&gt;* root);    //求叶子结点数 void mainpp();    //输出菜单</pre>
界面设计和使用说明	以 c++ 为开发语言，在 Visual Studio 2017 编译器上实现。 界面上显示执行简单测试程序后的结果。
调试分析	

### 8-1

abc##d##ef###

abcdef

cbdafe

cdbfea

abecdf

e

f

a

d

b

c

请按任意键继续. . .

### 8-2

abc##d##ef###

3

6

3

e

f

a

d

b

c

c

b

d

a

f

e

请按任意键继续. . .

### 8-3

```
abc##d##ef###
```

```
push a
push b
push c
pop
c
pop
b
push d
pop
d
pop
a
push e
push f
pop
f
pop
e
```

请按任意键继续. . .

总结:

二叉树的建立、遍历以及求结点数、深度、叶子结点数过程中运用了大量的递归,编写程序时十分容易出错,同时调试也有困难。

重要算法:

```
void CreateBiTree(T space, T fin) //建立二叉树
```

```
{
    SqQueue<BiTNode<T>*> Q;
    T x;
    BiTNode<T>* cur;
    cin >> x;
    if (x != space && x != fin)
    {
        root = new BiTNode<T>(x, NULL, NULL);
        Q.Enqueue(root);
        while (!Q.IsEmpty())
        {
            Q.DeQueue(cur);
            cin >> x;
            if (x == fin)
                return;
            if (x == space)
                cur->lchild = NULL;
            else
            {
                cur->lchild = new BiTNode<T>(x, NULL, NULL);
            }
        }
    }
}
```

心得体会

```

        Q.Enqueue(cur->lchild);
    }
    cin >> x;
    if (x == fin)
        return;
    if (x == space)
        cur->rchild = NULL;
    else
    {
        cur->rchild = new BiTNode<T>(x, NULL, NULL);
        Q.Enqueue(cur->rchild);
    }
}
}
}

```

此函数的作用根据给定的判断输入空结点和判断输入结束的标识，根据先序输入建立二叉树，其中运用了队列。