

天津大学

数据结构上机实验报告

1. 题目:二叉树的双序遍历(Double-order traversal)

学生姓名	<u>王雨朦</u>
学生学号	<u>2016229082</u>
学院名称	<u>国际工程师学院</u>
专业	<u>计算机</u>
时间	<u>2016/11/24</u>

目 录

第一章 需求分析	1
1.1 原题表述	1
1.2 解决方案	1
第二章 概要设计	1
2.1 抽象数据类型	2
第三章 详细设计	3
3.1 程序代码	3
第四章 调试分析	4
4.1 调试过程	4
第五章 测试结果	5
5.1 测试过程	5

第一章 需求分析

1.1 原题表述

双序遍历是指：对于二叉树的每一个结点来说，先访问这个结点，再按双序遍历它的左子树，然后再一次访问这个结点，接下来按双序遍历它的右子树。试写出执行这种双序遍历的算法。

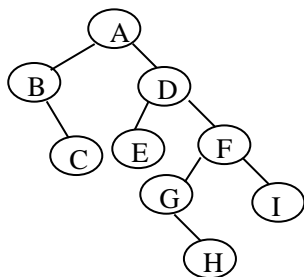
解决方案要求：

输入参数：创建好的一棵二叉树

输出参数：双序遍历结果

参考样例：

例如如果建立如下一个二叉树，则双序遍历结果应该为 ABBCADEDFGGHFI



- 1) 对先序遍历进行加工，访问根节点后，递归左子树；
- 2) 再访问根节点，递归右子树；
- 3) 输出遍历序列，完成遍历。

第二章 概要设计

2.1 抽象数据类型

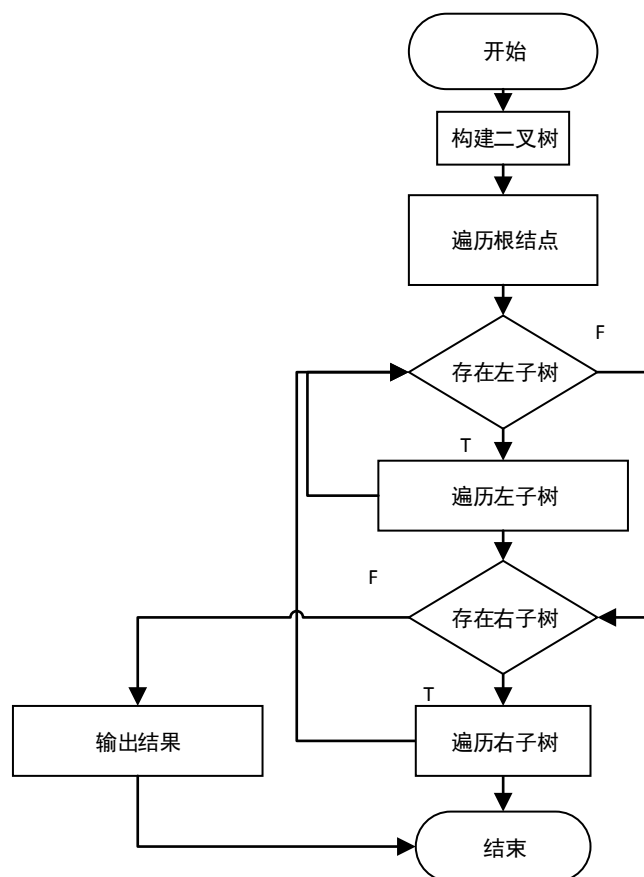
二叉树

```
typedef struct BinTreeNode
{
    char data;
    BinTreeNode *lchild, *rchild;
}*BinTree;
```

```
void CreateBinTree(BinTree & t)
```

```
void DoubleTraverse(BinTree t)
```

2.2 算法



第三章 详细设计

3.1 程序代码

```
#include <iostream>
using namespace std;
typedef struct BinTreeNode{
    char data;
    BinTreeNode *lchild, *rchild;
}*BinTree;
void CreateBinTree(BinTree & t){
    char d;
    cin >> d;
    t = new BinTreeNode;
    if(d!='#')
    {
        t->data = d;
        CreateBinTree(t->lchild);
        CreateBinTree(t->rchild);
    }
    else
        t=NULL;
}
void DoubleTraverse(BinTree t){
    if(t==NULL)
        return;
    else if(t->lchild==NULL&& t->rchild==NULL)
        cout << t->data;
    else {
        cout << t->data;
        DoubleTraverse(t->lchild);
        cout << t->data;
        DoubleTraverse(t->rchild);
    }
}
int main(){
    BinTree t;
    CreateBinTree(t);
    DoubleTraverse(t);
    return 0;
}
```

第四章 调试分析

4.1 调试过程

Bug 名称	逻辑错误
Bug 描述	没有理解清楚题意，对根节点进行了连续两次遍历
Bug 原因	不能简单粗暴的遍历两次
Bug 解决方案	在遍历左子树和右子树前，都对根节点进行一次遍历
Bug 总结	想清楚再写代码

第五章 测试结果

5.1 测试过程

测试编号	1
测试对象	双序遍历功能
测试输入参数	AB#C##DE##FG#H##I##
测试步骤	先输入二叉树,看其是否输出双序结果
测试预期结果	ABBCADEDFGGHFI
测试输出结果	ABBCADEDFGGHFI
测试分析	程序正确

```
AB#C##DE##FG#H##I##
ABBCADEDFGGHFI
Process returned 0 (0x0)   execution time : 40.779 s
Press any key to continue.
```