

Lab2 二叉树实验报告

一、功能简介

1. 建立二叉树

实现了尽可能多的二叉树的建立，支持**多棵二叉树的合并**，支持用户对每棵二叉树**独立命名与调用**。

2. 建立节点

支持用户对节点的尽可能多的操作，包括但不限于**建立左节点、建立右节点、查看左节点、查看右节点、查看兄弟节点、查看父亲节点、返回父亲节点**等功能，并且在首次到达节点时会显示尽可能多的节点信息。

3. 查找二叉树

实现了两种查找方式。一种是按**二叉树的命名**查找，一种是按**二叉树的编号**查找。可以同时满足普通用户与管理者的查找需求。

4. 遍历二叉树

实现了**四种遍历方式与其中代码实现方式**。其中**先序遍历、中序遍历、后序遍历**各自提供了递归版与迭代版两种实现方式。**层次遍历**支持一种代码实现方式。尽可能满足用户对不同实现的要求。

5. 其他功能

实现了独立的**查看所有叶节点数与查看兄弟节点**等功能，方便用户频繁的独立调用需求。

二、具体实现方式与测试样例

对于连续输入输出，下面所有测试样例均采用红进黑出的方式。

建立二叉树

- 建立二叉树

- 实现方式

建立结构体，内含一个string对象与一个二叉树类的地址。二叉树类包含二叉树节点累，大部分操作都由二叉树调用节点的函数实现。通过用户输入二叉树的命名后赋值给string，然后读入二叉树根节点的初始化值，提示建立成功。

- 测试样例

- 输入与输出:

欢迎您使用建立二叉树功能

请输入您想建立的二叉树的名字:

tree

请输入您的二叉树的根节点的值:

1

提示:您已成功建立二叉树!执行返回操作

建立节点

在选择此功能后，会提示用户先进行二叉树的选择，然后报告根节点的信息，之后遍历到新的节点之后都会先行展示该节点的所有信息。

- 输入与输出:

欢迎使用建立节点功能!

请选择您想建立节点的二叉树

1.按名称选择

2.按编号查找

请输入您的选择:

1

请输入您所查找的二叉树名称:

tree

您所查找的二叉树名称为:tree

该二叉树的规模为:0

提示:查找成功!

当前节点高度为:0

当前节点信息为:1

当前节点没有左孩子.

当前节点没有右孩子.

1.建立左节点

- 实现思路

通过检查当前节点的左右节点地址是否为**NULL**确定有无左节点，有的话提示操作非法，没有的话利用**构造函数**对该节点进行建立左节点的操作。

- 测试样例一

- 输入与输出:

请输入您想建立的左节点的信息:

2

提示:建立成功!

- 测试样例二

- 输入与输出

请输入您想建立的左节点的信息:

2

警告:检测到非法操作!执行重置操作

2. 建立右节点

- 实现思路

与建立左节点基本一致。

- 测试样例一

- 输入与输出:

请输入您想建立的右节点的信息:

3

提示:建立成功

- 测试样例二

- 输入与输出:

请输入您想建立的右节点的信息:

3

警告:检测到非法操作!执行重置操作

3. 查看左节点

- 实现思路
通过二叉树节点类中的函数，调用temp_x->lChild中的height与data对象，获取左节点的基本信息，并使temp_x转换为左节点的地址，实现控制权的转换与节点的转移。如果当前节点没有左节点则提示非法操作。
 - 测试样例一
 - 输出:
当前节点左节点的高度为:0
当前节点左节点的信息为:2
提示:查看成功!执行重置操作
提示:当前控制权已转移到左节点.
 - 测试样例二
 - 输出:
警告:检测到非法操作!执行重置操作
-

4. 查看右节点

- 实现思路
同查看左节点
 - 测试样例一
 - 输出:
当前节点右节点的高度为:0
当前节点右节点的信息为:3
提示:查看成功!执行重置操作
提示:当前控制权已转移到右节点.
 - 测试样例二
 - 输出: 警告:检测到非法操作!执行重置操作
-

5. 查看兄弟节点

- 实现思路
通过对二叉树节点类中的parent的判断实现。首先检查该节点是否为根节点，如果是，提示用户该节点是根节点。如果不是根节点，就判断parent的lChild与rChild是否为空，然后确定这个节点是父亲节点的左节点还是右节点，然后输出对应兄弟节点的信息。
 - 测试样例一
 - 输出:
警告:查询失败!
当前节点为根节点
 - 测试样例二
 - 输出:
当前节点无兄弟节点
 - 测试样例三
 - 输出:
当前节点有兄弟节点,且兄弟节点的值为: 3
-

6. 查看父亲节点

- 实现思路
调用parent对象即可，唯一需要注意的情况是根节点。对其进行**特判**即可。
 - 测试样例一
 - 输出:
当前节点父亲节点的高度为:1
当前节点父亲节点的信息为:1
提示:查看成功!
 - 测试样例二
 - 输出:
警告:检测到格式错误!执行重置操作
-

7. 返回父亲节点

- 实现思路
调用parent并将地址转给该父亲节点即可。
 - 测试样例一
 - 输出:
提示:返回成功!
 - 测试样例二
 - 输出:
提示:检测到格式错误!执行重置操作
-

查找二叉树

- 实现方式
实现了两种查找方式。得益于结构体的实现，二叉树的各个信息都被储存了起来。利用名称查找，复杂度为 $O(n)$ ，遍历已储存的结构体，直到找到对应二叉树。按编号查找复杂度为 $O(1)$ ，相当于循秩查找，直接按下角标提取结构体即可。如果无用户输入的名字或编号超出实际范围，提示查找失败，执行重置操作。
- 测试样例一
 - 输入与输出:
欢迎使用查找二叉树功能!
请选择您查找的方式
1.按名称查找
2.按顺序查找
请输入您的选择:
1
请输入您所查找的二叉树名称:
tree
您所查找的二叉树名称为:tree
该二叉树的规模为:3
提示:查找成功!执行返回操作
- 测试样例二

- **输入与输出:**
欢迎使用查找二叉树功能!
请选择您查找的方式
1.按名称查找
2.按顺序查找
请输入您的选择:
2
请输入您所查找的二叉树编号:
1
您所查找的二叉树名称为:tree
该二叉树的规模为:3
提示:查找成功!执行返回操作
-

遍历二叉树

1. 先序遍历

- 实现思路 递归版每到达一个节点，就执行visit，随后进行左右的递归。迭代版思路为建立**辅助栈**，遇到的右节点压栈，然后visit该左节点，最后出栈即可。
- 测试样例
 - **输出:**
data:1
data:2
data:4
data:5
data:3
data:6
data:7

2. 中序遍历

- 实现思路 递归版每到达一个节点，先进行左节点的递归，再执行visit，随后进行右节点的递归。迭代版思路为建立**辅助栈**，将当前节点入栈后随即向左侧分支深入，然后出栈并visit即可。
- 测试样例
 - **输出:**
data:4
data:2
data:5
data:1
data:6
data:3
data:7

3. 后序遍历

- 实现思路 递归版每到达一个节点，先进行左右的递归，再执行visit。迭代版思路为建立**辅助栈**，将沿途所遇节点依次入栈，尽可能向左进行压栈操作——将右节点先压栈，再进行左节点压栈，然后弹出栈顶并visit即可。
- 测试样例
 - **输出:**
data:4

data:5
data:2
data:6
data:7
data:3
data:1

4. 层次遍历

- 实现思路 建立**辅助队列**，根节点入栈，然后取出队首元素进行visit，然后左孩子，右孩子依次入队，直到队列为空即可。
- 测试样例
 - 输出:
data:1
data:2
data:3
data:4
data:5
data:6
data:7

其他功能

1. 叶结点数

- 实现方式 基于遍历的代码实现而成，对每个遍历到的节点，建立全局变量记录之，最后输出访问次数即可。
- 测试样例
 - 输出
所有的叶节点数为:7
提示:计数成功!执行返回操作

三、优势与不足

优势

- 本二叉树优势主要在于**功能全面**，满足了用户对二叉树的绝大部分要求。无论是对节点的各项操作还是遍历查找的多样性，二叉树都体现出与同类产品相比卓越的性能。
- 二叉树依旧继承了本系列**用户友好**的传统特点，每一步都有详尽的使用指导与错误提示，简洁优美的界面可以给用户提供较好的体验。

不足

- 二叉树在**部分高级功能**尚有待完善，例如查看所有祖宗节点，查看所有子孙节点，二叉树的可视化等方面依旧有不足，对高端用户的需求恐怕难以满足。
- 存在可能的bug

四、总结

1. 收获

本次Lab历时2天，在二叉树的代码方面通过一个一个字母的逐个输入，获得了对二叉树深刻的理解，掌握了对二叉树基本的应用方法，为以后的学习打下了坚实的基础。

2. 参考资料

《数据结构(c++语言版)》邓俊辉编著 清华大学出版社

3. 项目地址

本次Lab所有文件已上传至<https://github.com/rucerchui/Sophomore/tree/main/Lab2>

欢迎您的使用！
