

数据结构 2022秋 Lab4

吴逸昕 19302010013

说明

本次 Lab 考察二叉树的前序、中序、后序、层序遍历。

提交

截止日期 **2022年10月23日 23:59:59**，只需提交 `BinaryTree.cpp` 文件，将代码打包为 zip，命名为 学号_姓名_Lab4.zip，提交于 elearning。

评分

将基于测试用例通过情况评分；若存在编译失败、运行时错误等问题，最多仅可得 3 分，**请提交前务必确保能够通过编译，且不会产生无效的内存引用等问题**。请勿修改已给出的函数接口、类名、文件名等，若因此导致测试用例无法通过则后果自负。

任务

1. 二叉树的 4 种遍历

在 `BinaryTree.cpp` 中，实现二叉树的 **前序、中序、后序、层序** 4 种遍历方式，并通过 `BinaryTreeTest.cpp` 测试。可运用已学习的数据结构，并自由添加其他辅助函数。

2. 二叉树中序遍历的非递归实现

在 `BinaryTree.cpp` 中，通过 **循环**，实现二叉树的 **中序** 遍历。可运用已学习的数据结构，并自由添加其他辅助函数。

```
#include<vector>
#include<stack>
#include<queue>
#include "BinaryNode.cpp"

using namespace std;

template<typename T> class BinaryTree {
private:
    BinaryNode<T> *root;
    // You may define some helper functions below:

public:
    /**
     * @brief Creates a binary tree with a binary node as its root.
     * */
    explicit BinaryTree(BinaryNode<T> *root) {
```

```

        this->root = root;
    }

    /**
     * @brief Preorder traverse, recursive.
     * @return Traverse result in a vector.
     * */
    vector<T> preOrder() {
        // TODO
        return vector<T>();
    }

    /**
     * @brief Inorder traverse, recursive.
     * @return Traverse result in a vector.
     * */
    vector<T> inOrder() {
        // TODO
        return vector<T>();
    }

    /**
     * @brief Postorder traverse, recursive.
     * @return Traverse result in a vector.
     * */
    vector<T> postOrder() {
        // TODO
        return vector<T>();
    }

    /**
     * @brief Level Order traverse, recursive.
     * @return Traverse result in a vector.
     * */
    vector<T> levelOrder() {
        // TODO
        return vector<T>();
    }

    /**
     * @brief Inorder traverse, NON-RECURSIVE.
     * @return Traverse result in a vector.
     * */
    vector<T> inOrderNonRecur() {
        // TODO
        return vector<T>();
    }
};

```

提示

一些可能会使用的标准库头文件已添加，可自行了解标准库中定义的相关数据结构的具体用法并直接使用，无需导入 Lab2 中自己的实现。可参考以下文档：

Stack 类：<https://learn.microsoft.com/zh-cn/cpp/standard-library/stack-class>

Queue 类：<https://learn.microsoft.com/zh-cn/cpp/standard-library/queue-class>