**判断二叉树的异同**

1. **实验目的和要求**

设计一个函数，判别两颗二叉树是否相同。所谓两棵树相同是指树的形状及对应的结点的值相同。设二叉树以二叉链表的方式储存。

**二、实验内容**

1. 实验准备
   1. 理论知识介绍
      1. 树的定义:
         1. 树形结构的特点是一个数据就够可以有很多个直接后继，但只有一个直接前驱，而线性结构中的每个数据元素至多只有一个直接前驱和一个直接后继。
      2. 二叉树
         1. 二叉树是结点的有限集合，它或者为空，或者由一个根结点及两颗互不相交的左、右子树构成，而其左、右子树又都是二叉树。
         2. 如果一颗二叉树中任意一层的结点个数都达到了最大值，那么这颗二叉树称为满二叉树或丰满树。
         3. 完全二叉树是在满二叉树的最底层自右至左依次（不能跳过任何一个结点）去掉若干个结点。
         4. 二叉树的链接实现
            1. 二叉树的主要储存方式是链接储存，用链接储存结构储存二叉树是很自然的，二叉树的链接储存就是用指针指出父子关系。大多数树的应用只需要找结点的儿子，因此每个结点只需要指出它的子结点储存在哪里。但也有一些应用需要找结点的父亲，因此还必须指出它的父结点储存在哪里。这样就星辰了两种链接储存结构：标准储存结构和广义的标准储存结构。
            2. 标准储存结构也称为二叉链表。在二叉链表中，每个储存结点由三个字段组成，储存结构数据元素值的数据字段以及指向左、右儿子的指针字段。data储存二叉树节点中的数据元素，left和right存放左、右儿子的地址。left和right又是也被称为左、右指针。
2. 实验项目
   1. 分析
      1. 由于判断树是否相等主要就是判断树中的数据元素（即树包含的结点结构）是否相等，所以我们可以先构造一个判断两个结点结构是否相等的函数模板。
      2. 为了判断两个结点结构是否相等。首先我们要判断这两个结点结构的根结点中的元素是否相同，然后判断根结点的两个儿子是否相同，判断两个儿子子否相同的方法应该是和判断根结点相同的方法一样的，所以我们应该用递归函数的方法来判断。
   2. 方案
      1. 构造一个结点类。
      2. 利用结点类构造一个树的类。
      3. 构造一个函数模板判断两个结点类是否相同，需要判断的是这个结点的数据元素值和两个儿子结点是否相同。
      4. 利用上述的函数模板构造一个判断二叉树是否相同的函数模板。
      5. 测试函数模板
   3. 测试数据

int main() {

Node<int> node1(1);

Node<int> node2(2);

Node<int> node3(3);

Node<int> \* p1 = &node1;

Node<int> \* p2 = &node2;

Node<int> \* p3 = &node3;

Node<int> node4(4, p1, p2);

Node<int> node5(4, p1, p2);

Node<int> node6(6, p2, p3);

BinaryTree<int> bt4(node4);

BinaryTree<int> bt5(node5);

BinaryTree<int> bt6(node6);

cout << TreesAreSame(&bt4, &bt5) << endl;

cout << TreesAreSame(&bt5, &bt6) << endl;

return 0;

}

* 1. 测试结果

1

0

* 1. 出错情况

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Severity | Code | Description | Project | File | Line | Suppression State |
| Error | C2664 | 'bool NodesAreSame<Type>(Node<Type> \*,Node<Type> \*)': cannot convert argument 1 from 'int' to 'Node<int> \*' | BinaryTree | c:\users\93137\source\repos\binarytree\binarytree\binarytree.cpp | 42 |  |

这个错误是由于在构造树的类的时候没有根据实际的情况将构造函数的返回值修改成对应的值造成的错误。

还有一个错误是由于在构造二叉树的判断相同的函数模板的时候，需要用到判断结点是否相同的函数模板的时候没有加入<Type>从而引发的编译器无法判断结点函数模板参数类型的错误，但是这个错误在第二次编译的时候无法重现（可能是编译器在正确编译一次之后储存了该类型的信息）。

**三、实验小结**

1. 重点

二叉树的实现以及二叉树的应用。

1. 难点

如何判断两个二叉树是否相同，需要利用递归函数。

1. 解决方法

判断二叉树是否相同其实就是判断二叉树中的结点的结构是否相同。要判断结点的结构是否相同就需要先判断根节点的元素值是否相同，以及根结点的两个子结点是否相同，子结点的判断利用递归的方法。

1. 收获、体会

在运用函数模板和类模板的时候经常由于没有在特定的位置增添模板类型的尖括号而出错，主要是由于对模板的不熟悉。对于结构有相似关系的类型我们经常可以用到递归函数的方法来简化问题的过程。