

输入两颗树A, B, 判断B是不是A的子结构

```
public class B26 {  
    public static class TreeNode {  
        int val;  
        public TreeNode left;  
        public TreeNode right;  
  
        public TreeNode(int val) {  
            this.val = val;  
        }  
    }  
    /*  
    * 这是一个递归解法, 判断B是不是A的子结构, 然后再判断B是不是A的左子树的子结构, B是不是A的右子树的子结构。  
    */  
    public static boolean hasSubTree(TreeNode root1, TreeNode root2) {  
        if (root1 == null || root2 == null)  
            return false;  
        return isTheSameTree(root1, root2) || hasSubTree(root1.left, root2) || hasSubTree(root1.right, root2);  
    }  
  
    public static boolean isTheSameTree(TreeNode root1, TreeNode root2) {  
        if (root2 == null) {  
            return true;  
        }  
        if (root1 == null && root2 != null) {  
            return false;  
        }  
  
        return root1.val == root2.val && isTheSameTree(root1.left, root2.left) && isTheSameTree(root1.right, root2.right);  
    }  
}
```