# 树的子结构

## 题目

[牛客网](https://www.nowcoder.com/practice/6e196c44c7004d15b1610b9afca8bd88?tpId=13&tqId=11170&rp=1&ru=%2Fta%2Fcoding-interviews&qru=%2Fta%2Fcoding-interviews%2Fquestion-ranking&tPage=1)

输入两棵二叉树A，B，判断B是不是A的子结构。（ps：我们约定空树不是任意一个树的子结构）

## 解题思路

1. 遍历查找相等根节点
2. 通过递归查找当前根节点下是否包含子树 root2

public boolean HasSubtree(TreeNode root1, TreeNode root2) {
 if (root2 == null) {
 return false;
 }

 LinkedList<TreeNode> pipeline = new LinkedList<>();
 pipeline.addLast(root1);

 while (!pipeline.isEmpty()) {
 TreeNode node = pipeline.pop();
 if (node == null) {
 continue;
 }

 pipeline.addLast(node.left);
 pipeline.addLast(node.right);

 if (node.val == root2.val && isSub(node, root2)) {
 return true;
 }
 }

 return false;
}

private boolean isSub(TreeNode root1, TreeNode root2) {
 if (root1 == null && root2 == null) {
 return true;
 }

 if (root1 == null) {
 return false;
 }

 if (root2 == null) {
 return true;
 }

 if (root1.val == root2.val) {
 return isSub(root1.left, root2.left) && isSub(root1.right, root2.right);
 } else {
 return false;
 }

}