# 题目

检查子树。你有两棵非常大的二叉树：T1，有几万个节点；T2，有几万个节点。设计一个算法，判断 T2 是否为 T1 的子树。

如果 T1 有这么一个节点 n，其子树与 T2 一模一样，则 T2 为 T1 的子树，也就是说，从节点 n 处把树砍断，得到的树与 T2 完全相同。

注意：此题相对书上原题略有改动。

示例1:

输入：t1 = [1, 2, 3], t2 = [2]

输出：true

示例2:

输入：t1 = [1, null, 2, 4], t2 = [3, 2]

输出：false

提示：

树的节点数目范围为[0, 20000]。

# 分析

/\*\*

 \* Definition for a binary tree node.

 \* struct TreeNode {

 \*     int val;

 \*     TreeNode \*left;

 \*     TreeNode \*right;

 \*     TreeNode(int x) : val(x), left(NULL), right(NULL) {}

 \* };

 \*/

class Solution {

public:

    bool checkSubTree(TreeNode\* t1, TreeNode\* t2) {

        if(!t1) return false;

        return checkSubTree(t1->left,t2) || isSame(t1,t2) || checkSubTree(t1->right,t2);

    }

    bool isSame(TreeNode \*s,TreeNode \*t)

    {

        if(nullptr==s && nullptr==t)    return true;

        if(nullptr==s || nullptr==t)    return false;

        return s->val==t->val && isSame(s->left,t->left) && isSame(s->right,t->right);

    }

};