# 题目

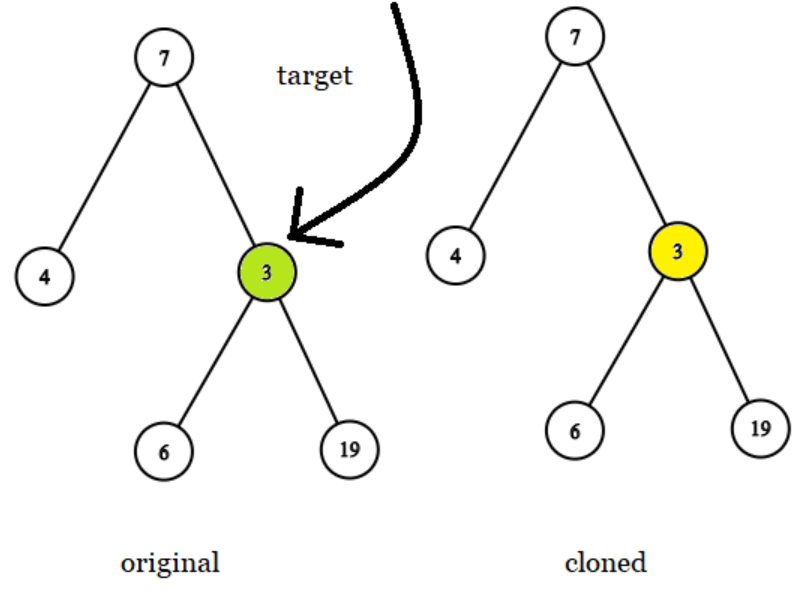
给你两棵二叉树，原始树original和克隆树cloned，以及一个位于原始树original中的目标节点target。

其中，克隆树cloned是原始树original的一个副本。

请找出在树cloned中，与target相同的节点，并返回对该节点的引用（在C/C++等有指针的语言中返回节点指针，其他语言返回节点本身）。

注意：你不能对两棵二叉树，以及target节点进行更改。只能返回对克隆树cloned中已有的节点的引用。

示例 1:

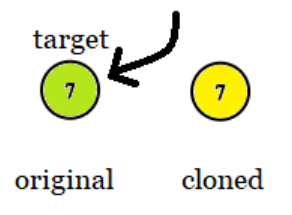


输入: tree = [7,4,3,null,null,6,19], target = 3

输出: 3

解释: 上图画出了树original和cloned。target节点在树original中，用绿色标记。答案是树cloned中的黄颜色的节点（其他示例类似）。

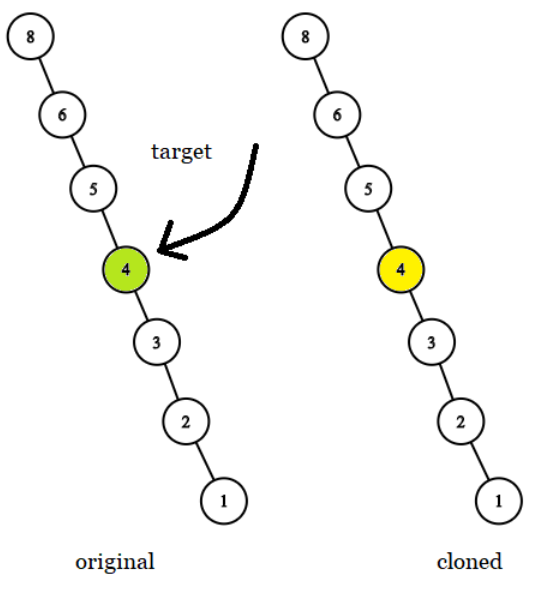
示例 2:



输入: tree = [7], target = 7

输出: 7

示例 3:



输入: tree = [8,null,6,null,5,null,4,null,3,null,2,null,1], target = 4

输出: 4

提示：

树中节点的数量范围为[1, 104] 。

同一棵树中，没有值相同的节点。

target节点是树original中的一个节点，并且不会是null。

进阶：如果树中允许出现值相同的节点，将如何解答？

# 分析

class Solution {

public:

TreeNode\* getTargetCopy(TreeNode\* original, TreeNode\* cloned, TreeNode\* target) {

if (original == target) return cloned;

if (original == nullptr || cloned == nullptr) return nullptr;

TreeNode \*left = getTargetCopy(original->left, cloned->left, target);

TreeNode \*right = getTargetCopy(original->right, cloned->right, target);

return left == nullptr ? right : left;

}

};