

LeetCode: [1325. 删除给定值的叶子节点](#) 题解:

给你一棵以 `root` 为根的二叉树和一个整数 `target`，请你删除所有值为 `target` 的叶子节点。

注意，一旦删除值为 `target` 的叶子节点，它的父节点就可能变成叶子节点；如果新叶子节点的值恰好也是 `target`，那么这个节点也应该被删除。

也就是说，你需要重复此过程直到不能继续删除。

输入: `root = [1,2,3,2,null,2,4]`, `target = 2`

输出: `[1,null,3,null,4]`

解释:

上面左边的图中，绿色节点为叶子节点，且它们的值与 `target` 相同（同为 2），它们会被删除，得到中间的图。

有一个新的节点变成了叶子节点且它的值与 `target` 相同，所以将再次进行删除，从而得到最右边的图。

输入: `root = [1,3,3,3,2]`, `target = 3`

输出: `[1,3,null,null,2]`

输入: `root = [1,2,null,2,null,2]`, `target = 2`

输出: `[1]`

解释: 每一步都删除一个绿色的叶子节点（值为 2）。

输入: `root = [1,1,1]`, `target = 1`

输出: `[]`

输入: `root = [1,2,3]`, `target = 1`

输出: `[1,2,3]`

来源: 力扣 (LeetCode)

链接: <https://leetcode-cn.com/problems/delete-leaves-with-a-given-value>

著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。

想查看图片就点一下链接吧

思路简介:

1. 递归好了，比较简洁
2. 递归出口: 结点为 `NULL`
3. 先清理左子树与`target`相同的结点
4. 再清理右子树与`target`相同的结点
5. 如果`root`结点也是叶结点且与`target`的值相同，返回 `NULL`

show code:

```
/**
 * 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:
    TreeNode* removeLeafNodes(TreeNode* root, int target) {
        if(root==NULL) return NULL;
        root->left=removeLeafNodes(root->left,target);
        root->right=removeLeafNodes(root->right,target);
        if(root->left==NULL&&root->right==NULL&&root->val==target)
            return NULL;
        return root;
    }
};
```