

二叉树的堂兄弟节点

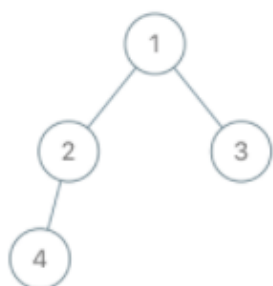
在二叉树中，根节点位于深度 0 处，每个深度为 k 的节点的子节点位于深度 $k+1$ 处。

如果二叉树的两个节点深度相同，但 **父节点不同**，则它们是一对**堂兄弟节点**。

我们给出了具有唯一值的二叉树的根节点 `root`，以及树中两个不同节点的值 x 和 y 。

只有与值 x 和 y 对应的节点是堂兄弟节点时，才返回 `true`。否则，返回 `false`。

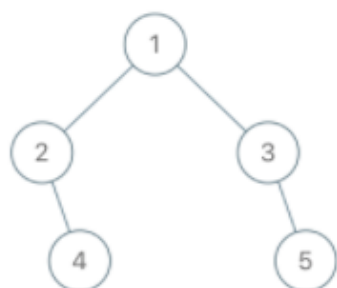
示例 1:



输入: `root = [1,2,3,4]`, $x = 4$, $y = 3$

输出: `false`

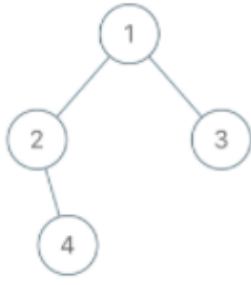
示例 2:



输入: `root = [1,2,3,null,4,null,5]`, $x = 5$, $y = 4$

输出: `true`

示例 3:



输入: root = [1,2,3,null,4], x = 2, y = 3

输出: false

```
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     TreeNode *left;
 *     TreeNode *right;
 *     TreeNode() : val(0), left(nullptr), right(nullptr) {}
 *     TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}
 *     TreeNode(int x, TreeNode *left, TreeNode *right) : val(x), left(left
), right(right) {}
 * };
 */
class Solution {
    int depth1,depth2,data1,data2;
public:
    void dfs(TreeNode* root,int depth,int data,int num1,int num2)
    {
        if(root==nullptr)
        {
            return;
        }
        if(root->val==num1)//如果当前节点找到 x 记录值和深度
        {
            depth1=depth;
            data1=data;
        }
        else if(root->val==num2)//如果当前节点找到 y 记录值和深度
        {
            depth2=depth;
            data2=data;
        }
        dfs(root->left,++depth,root->val,num1,num2); //找左边
```

```
    depth--;//回退到上一层
    dfs(root->right,++depth,root->val,num1,num2);//找右边
}
bool isCousins(TreeNode* root, int x, int y) {
    dfs(root,0,root->val,x,y);
    return depth1==depth2&&data1!=data2;
}
};
```