Problem 993: Cousins in Binary Tree

Problem Information

Difficulty: Easy
Acceptance Rate: 58.79%
Paid Only: No
Tags: Tree, Depth-First Search, Breadth-First Search, Binary Tree

Problem Description

Given the `root` of a binary tree with unique values and the values of two different nodes of the tree `x` and `y`, return `true` _if the nodes corresponding to the values_`x` _and_`y` _in the tree are**cousins** , or _`false` _otherwise._

Two nodes of a binary tree are **cousins** if they have the same depth with different parents.

Note that in a binary tree, the root node is at the depth `0`, and children of each depth `k` node are at the depth `k + 1`.

Example 1:

Input: root = [1,2,3,4], x = 4, y = 3 **Output:** false

Example 2:

Input: root = [1,2,3,null,4,null,5], x = 5, y = 4 **Output:** true

Example 3:

Input: root = [1,2,3,null,4], x = 2, y = 3 **Output:** false

Constraints:

* The number of nodes in the tree is in the range `[2, 100]`. * `1 <= Node.val <= 100` * Each node has a **unique** value. * `x != y` * `x` and `y` are exist in the tree.

Code Snippets

C++:

```
/**
* 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 {
public:
bool isCousins(TreeNode* root, int x, int y) {

}
};
```

Java:

```
/**
* Definition for a binary tree node.
* public class TreeNode {
* int val;
* TreeNode left;
* TreeNode right;
* TreeNode() {}
* TreeNode(int val) { this.val = val; }
* TreeNode(int val, TreeNode left, TreeNode right) {
```

```
* this.val = val;
* this.left = left;
* this.right = right;
* }
* }
*/
class Solution {
public boolean isCousins(TreeNode root, int x, int y) {

}
}
```

Python3:

```
# Definition for a binary tree node.
# class TreeNode:
# def __init__(self, val=0, left=None, right=None):
# self.val = val
# self.left = left
# self.right = right
class Solution:
def isCousins(self, root: Optional[TreeNode], x: int, y: int) -> bool:
```