/\*In a binary tree, the root node is at depth 0, and children of each depth k node are at depth k+1.

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

We are given the root of a binary tree with unique values, and the values x and y of two different nodes in the tree.

Return true if and only if the nodes corresponding to the values x and y are cousins\*/

class Solution {

private int height(TreeNode root,int val){

if (root == null){

return Integer.MIN\_VALUE;

}

if(root.val == val)

return 0;

int left = 0, right = 0;

left=height(root.left,val);

right = height(root.right,val);

return Math.max(left,right)+1;

}

private TreeNode parent\_check(TreeNode root,int val){

if(root == null)

return null;

if(root.left != null && root.left.val == val)

return root;

if(root.right != null && root.right.val == val)

return root;

TreeNode left = parent\_check(root.left,val);

if (left!=null)

return left;

return parent\_check(root.right,val);

}

public boolean isCousins(TreeNode root, int x, int y) {

if(root == null)

return false;

TreeNode par\_x = parent\_check(root,x);

TreeNode par\_y = parent\_check(root,y);

if(par\_x!=par\_y){

int h\_x = height(root,x);

int h\_y = height(root,y);

if(h\_x == h\_y)

return true;

}

return false;