LeetCode

Given a binary tree, check whether it is a mirror of itself (ie, symmetric around its center).

/\*\*

\* 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 isSymmetric(TreeNode\* root) {

return isMirror(root,root);

}

bool isMirror(TreeNode\* root1,TreeNode\* root2)

{

if(root1==NULL && root2==NULL)

return true;

if(root1==NULL && root2!=NULL)

return false;

if(root1!=NULL && root2==NULL)

return false;

if(root1 && root2 && root1->val==root2->val)

return isMirror(root1->left,root2->right)&& isMirror(root1->right,root2->left);

return false;

}

};