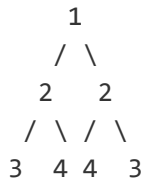


Problem

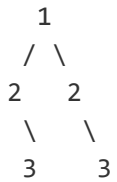
101. Symmetric Tree(Easy)

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

For example, this binary tree [1,2,2,3,4,4,3] is symmetric:



But the following [1,2,2,null,3,null,3] is not:



Note:

Bonus points if you could solve it both recursively and iteratively.

Solution

O(n) time, O(1) space

一道典型的递归求解的问题

[GitHub传送门](#)

```
class Solution
{
    bool check(TreeNode *node1, TreeNode *node2)
    {
        if (node1 == nullptr || node2 == nullptr)
            return node2 == nullptr && node1 == nullptr;
        if (node1->val == node2->val)
            return check(node1->left, node2->right) && check(node1->right, node2->left);
        return false;
    }

public:
    bool isSymmetric(TreeNode *root)
    {
        if (root == nullptr)
            return true;
        return check(root->left, root->right);
    }
};
```