

## 剑指28 对称的二叉树

想了半天。

如果一棵二叉树和它的镜像一样，那么它是对称的。

即，它的左子树和右子树，是对称的。

即 1.左子树和右子树根节点值相等。且

2.左子树的右子树 和 右子树的左子树对称。且

3.左子树的左子树 和 右子树的右子树对称。

```
1 // 我的答案1: 几乎双百。(其实是因为大家答案都一样)
2
3 /**
4  * Definition for a binary tree node.
5  * struct TreeNode {
6  *     int val;
7  *     TreeNode *left;
8  *     TreeNode *right;
9  *     TreeNode(int x) : val(x), left(NULL), right(NULL) {}
10 * };
11 */
12
13
14 class Solution {
15     bool ismirror(TreeNode* A, TreeNode* B) {
16         if (A == NULL && B == NULL) return true;
17         else if (A != NULL && B != NULL) {
18             if (A->val != B->val)
19                 return false;
20             else
21                 return ismirror(A->left, B->right) && ismirror(A->right, B->left);
22         }
23         else
24             return false;
25     }
26 public:
27     bool isSymmetric(TreeNode* root) {
28         if (root == NULL) return true;
29         return ismirror(root->left, root->right);
30     }
31 };
```

非递归方法:

妙啊

```
1 // 别人的非递归，层序遍历。
2 // 一开始想这样，但我自己没实现的。
3
4 bool isSymmetric(TreeNode* root) {
5     if (!root) return true;
```

```
6 queue<TreeNode*> q;
7 q.push(root->left);
8 q.push(root->right);
9 while(q.size() > 1) {
10     TreeNode* left = q.front();
11     q.pop();
12     TreeNode* right = q.front();
13     q.pop();
14     if (!left && !right) continue;
15     if (!left || !right)
16         return false;
17     if (left->val != right->val)
18         return false;
19     q.push(left->left);
20     q.push(right->right);
21     q.push(left->right);
22     q.push(right->left);
23 }
24
25 return true;
26 }
```