

## 第42题——对称的二叉树

请实现一个函数，用来判断一棵二叉树是不是对称的。如果一棵二叉树和它的镜像一样，那么它是对称的。

例如，二叉树 [1,2,2,3,4,4,3] 是对称的。

```
1  /
 2 2 /\
```

3 4 4 3 但是下面这个 [1,2,2,null,3,null,3] 则不是镜像对称的:

```
1  /
 2 2 \
 3 3
```

示例 1:

输入: root = [1,2,2,3,4,4,3]

输出: true

示例 2:

输入: root = [1,2,2,null,3,null,3]

输出: false

## 解题思路:

二叉树的题目一般都考察递归

这道题也是一样:

左右子树都为空,返回true

左子树或者右子树任意一个为空,或者左右子树的值不相等,返回false

```
/**
 * Definition for a binary tree node.
 * struct TreeNode {
```

```

*   int val;
*   TreeNode *left;
*   TreeNode *right;
*   TreeNode(int x) : val(x), left(NULL), right(NULL) {}
* };
*/
class Solution {
public:
    bool isSymmetric(TreeNode* root) {
        if(root==NULL)
        {
            return true;
        }
        return dfs(root->left,root->right); //递归左右子树
    }
    bool dfs(TreeNode* A,TreeNode* B)
    {
        if(A==NULL && B==NULL){
            return true;
        }
        if(A==NULL || B==NULL || A->val!=B->val){
            return false;
        }
        return dfs(A->left,B->right)&&dfs(A->right,B->left); //对称递归遍历
    }
};

```