/\*\*

\* Definition for a binary tree node.

\* struct TreeNode {

\* int val;

\* TreeNode \*left;

\* TreeNode \*right;

\* TreeNode(int x) : val(x), left(NULL), right(NULL) {}

\* };

\*/

/\*

我的解题思路：中树遍历之后，用数组存储大小，看是否成中心对称的数字集合

Dicussion里面：1.判断一个点的左右节点是否为空且是否value是否相等

问题：1.知道思路之后 不知道是在原函数里面使用递归 还是在新函数中，不知道用两个指针

2.错误：等号输入错误

\*/

class Solution {

public:

bool isSymmetric(TreeNode\* root) {

if(root==NULL)

return true; //easy to error

if(root->left!=NULL && root->right!=NULL)

return isEqual(root->left,root->right);

else if (root->left==NULL&& root->right==NULL)

return true;

else

return false;

}

private:

bool isEqual(TreeNode \*left , TreeNode \*right)

{

bool lflag,rflag,localflag;

if(left!=NULL && right!=NULL)

{

if(left->val==right->val)

{

localflag=true;

lflag=isEqual(left->left,right->right);

rflag=isEqual(left->right,right->left);

if(localflag && lflag && rflag)

return true;

else

return false;

}

else

return false;

}

if(left==NULL && right!=NULL)

return false;

if(left!=NULL && right==NULL)

return false;

if(left==NULL && right==NULL)

return true;

}

};

思考：自己思路复杂，另外衍生的是关于树遍历的代码书写