Symmetric Tree

原本思路：中序遍历完二叉树把结点值放到vector后，判断vector里的值是否关于中点值对称。代码如下，可是有问题？：

class Solution {

public:

bool isSymmetric(TreeNode\* root) {

if(root==NULL)

return true;

stack<TreeNode\*> tree;

vector<int> arr;

TreeNode \*p;

p=root;

while(p||tree.size()!=0)

{

if(p)

{

tree.push(p);

p=p->left;

}

else

{

p=tree.pop(tree);

arr.push\_back(p->val);

p = p->right;

}

}

int leg=arr.size();

if(leg%2==0)

return false;

int mid;

mid=leg/2;

for(int i=0;i<mid;i++)

{

if(arr[i]!=arr[leg-1-i])

{

return false;

break;

}

}

return true;

}

};

递归算法代码如下：

class Solution {

public:

bool isSymmetric(TreeNode\* root) {

if(root==NULL)

return true;

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

}

bool isSymmetric(TreeNode\* lroot,TreeNode\* rroot)

{

if(lroot==NULL&&rroot==NULL)

return true;

if((lroot==NULL&&rroot!=NULL)||(lroot!=NULL&&rroot==NULL))

return false;

return (isSymmetric(lroot->left,rroot->right)&&isSymmetric(lroot->right,rroot->left)&&(lroot->val==rroot->val));

}

};