一、题目说明

题目\101. Symmetric Tree

二、我的解答

这个题目不难，但是做错了2次。递归方法的代码：

```
class Solution{
    public:
        bool isMirror(TreeNode* t1,TreeNode*t2){
            if(t1==NULL && t2==NULL) return true;
            if(t1==NULL || t2==NULL) return false;
            return (t1->val==t2->val) && isMirror(t1->left,t2->right) && isMirror(t1->right,t2->left);
        }
        bool isSymmetric(TreeNode* root){
            return isMirror(root,root);
        }
};
```

性能：

```
Runtime: 8 ms, faster than 52.13% of C++ online submissions for Symmetric Tree.
Memory Usage: 14.7 MB, less than 88.14% of C++ online submissions for Symmetric Tree.
```

三、优化措施

非递归算法，需要用一个队列queue，。

```
class Solution{
    public:
        //iterative
        bool isSymmetric(TreeNode* root){
            if(root==NULL || (root->left==NULL && root->right==NULL)){
                return true;
            }
            queue<TreeNode*> q;
            //将左右子树入队列
            q.push(root->left);
            q.push(root->right);
            while(!q.empty()){
                //左子树出队列
                TreeNode* l = q.front();
                q.pop();
                //左子树出队列
                TreeNode* r = q.front();
                q.pop();
                if(l ==NULL && r==NULL){
                    continue;
                }
                if(l==NULL || r==NULL){
                    return false;
                }
                if(l->val != r->val){
                    return false;
                }

                //将左子树的“左子树”，右子树的“右子树”入队列
                q.push(l->left);
                q.push(r->right);

                //将左子树的“右子树”，右子树的“左子树”入队列
                q.push(l->right);
                q.push(r->left);
            }

            return true;
        }
};
```

```
Runtime: 4 ms, faster than 85.48% of C++ online submissions for Symmetric Tree.
Memory Usage: 14.7 MB, less than 91.53% of C++ online submissions for Symmetric
Tree.
```