116. Populating Next Right Pointers in Each Node

题目描述：https://leetcode.com/problems/populating-next-right-pointers-in-each-node/

> 把二叉树转成链表二叉树。
>
> ```
> struct TreeLinkNode {
>   TreeLinkNode *left;
>   TreeLinkNode *right;
>   TreeLinkNode *next;
> }
> ```
>
> 例如：
>
> ```
>     1
>    /  \
>   2    3
>  / \  / \
> 4  5  6  7
> ```
>
> After calling your function, the tree should look like:
>
> ```
>     1 -> NULL
>    /  \
>   2 -> 3 -> NULL
>  / \  / \
> 4->5->6->7 -> NULL
> ```
>
> ### 解题思路：
>
> 按层次遍历
>
> ### 代码非递归：

```
/**
 * Definition for binary tree with next pointer.
 * struct TreeLinkNode {
 *  int val;
 *  TreeLinkNode *left, *right, *next;
 *  TreeLinkNode(int x) : val(x), left(NULL), right(NULL), next(NULL) {}
 * };
 */
class Solution {
public:
    void connect(TreeLinkNode *root) {
        if(root == NULL) return ;
        queue<TreeLinkNode*> q;
        q.push(root);
        while(!q.empty()) {
            int l = q.size();
            TreeLinkNode* pre = q.front();
            if(pre->left != NULL) q.push(pre->left);
            if(pre->right != NULL) q.push(pre->right);
            q.pop();
            l--;
            TreeLinkNode* p = NULL;
            while(l > 0) {
                l--;
                p = q.front();
                q.pop();
                pre->next = p;
                if(p->left != NULL) q.push(p->left);
                if(p->right != NULL) q.push(p->right);
                pre = p;
            }
        }
        return ;
    }
};
```

代码 递归:

代码2递归:

```
class Solution {
public:
    void connect(TreeLinkNode *root) {
        connect(root, NULL);
    }
private:
    void connect(TreeLinkNode *root,TreeLinkNode *sibling) {
        if(root == NULL)
            return;
        else
            root->next = sibling;
        connect(root->left, root->right);
        if(sibling != NULL)
            connect(root->right, sibling->left);
        else
            connect(root->right, NULL);
    }
};
```