Populating Next Right Pointers in Each Node

Given a binary tree

```
struct TreeLinkNode {
  TreeLinkNode *left;
  TreeLinkNode *right;
  TreeLinkNode *next;
}
```

Populate each next pointer to point to its next right node. If there is no next right node, the next pointer should be set to `NULL`.

Initially, all next pointers are set to `NULL`.

Note:

- You may only use constant extra space.
- You may assume that it is a perfect binary tree (ie, all leaves are at the same level, and every parent has two children).

For example,
Given the following perfect binary tree,

```
     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
```

Solution 1

```
void connect(TreeLinkNode *root) {
    if (root == NULL) return;
    TreeLinkNode *pre = root;
    TreeLinkNode *cur = NULL;
    while(pre->left) {
        cur = pre;
        while(cur) {
            cur->left->next = cur->right;
            if(cur->next) cur->right->next = cur->next->left;
            cur = cur->next;
        }
        pre = pre->left;
    }
}
```

you need two additional pointer.

written by ragepyre original link here

Solution 2

```
void connect(TreeLinkNode *root) {
    if(!root)
        return;
    while(root -> left)
    {
        TreeLinkNode *p = root;
        while(p)
        {
            p -> left -> next = p -> right;
            if(p -> next)
                p -> right -> next = p -> next -> left;
            p = p -> next;
        }
        root = root -> left;
    }
}
```

written by Erudy original link here

Solution 3

```
public class Solution {
    public void connect(TreeLinkNode root) {
        TreeLinkNode level_start=root;
        while(level_start!=null){
            TreeLinkNode cur=level_start;
            while(cur!=null){
                if(cur.left!=null) cur.left.next=cur.right;
                if(cur.right!=null && cur.next!=null) cur.right.next=cur.next.lef
t;

                cur=cur.next;
            }
            level_start=level_start.left;
        }
    }
}
```

written by talent58 original link here