117. Populating Next Right Pointers in Each Node II ★

Question	Editorial Solution

My Submissions (/problems/populating-next-right-pointers-in-each-node-ii/submissions/)

Total Accepted: **73210**   Total Submissions: **220618**   Difficulty: **Hard**

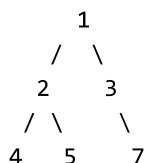Follow up for problem "*Populating Next Right Pointers in Each Node*".

What if the given tree could be any binary tree? Would your previous solution still work?

Note:

- You may only use constant extra space.

For example,

Given the following binary tree,

```
     1
    / \
   2   3
  / \   \
 4   5   7
```

After calling your function, the tree should look like:

```
     1 -> NULL
    / \
   2 -> 3 -> NULL
  / \     \
 4-> 5 -> 7 -> NULL
```

Subscribe (/subscribe/) to see which companies asked this question

Show Tags

Show Similar Problems

Have you met this question in a real interview? Yes No

Discuss (https://leetcode.com/discuss/questions/oj/populating-next-right-pointers-in-each-node-ii)

Pick One (/problems/random-one-question/)

C++   ▼   ⟳   </>

```
1  /**
2   * Definition for binary tree with next pointer.
3   * struct TreeLinkNode {
4   *  int val;
5   *  TreeLinkNode *left, *right, *next;
```

✉ Send Feedback (mailto:admin@leetcode.com?subject=Feedback)

```cpp
 6    *  TreeLinkNode(int x) : val(x), left(NULL), right(NULL), next(NULL) {}
 7    * };
 8    */
 9   class Solution {
10   public:
11       void connect(TreeLinkNode *root) {
12           while (root) {
13               TreeLinkNode *next = nullptr, *prev = nullptr;
14               for (; root; root = root->next) {
15                   if (!next) next = root->left ? root->left : root->right;

17                   if (root->left) {
18                       if (prev) prev->next = root->left;
19                       prev = root->left;
20                   }
21                   if (root->right) {
22                       if (prev) prev->next = root->right;
23                       prev = root->right;
24                   }
25               }
26               root = next;
27           }
28       }
```

Notes

Submit Solution

Privacy

✉ Send Feedback (mailto:admin@leetcode.com?subject=Feedback)