117. Populating Next Right Pointers in Each Node II

117. Populating Next Right Pointers in Each Node II

- **Depth-first Search** + Tree

Description

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.

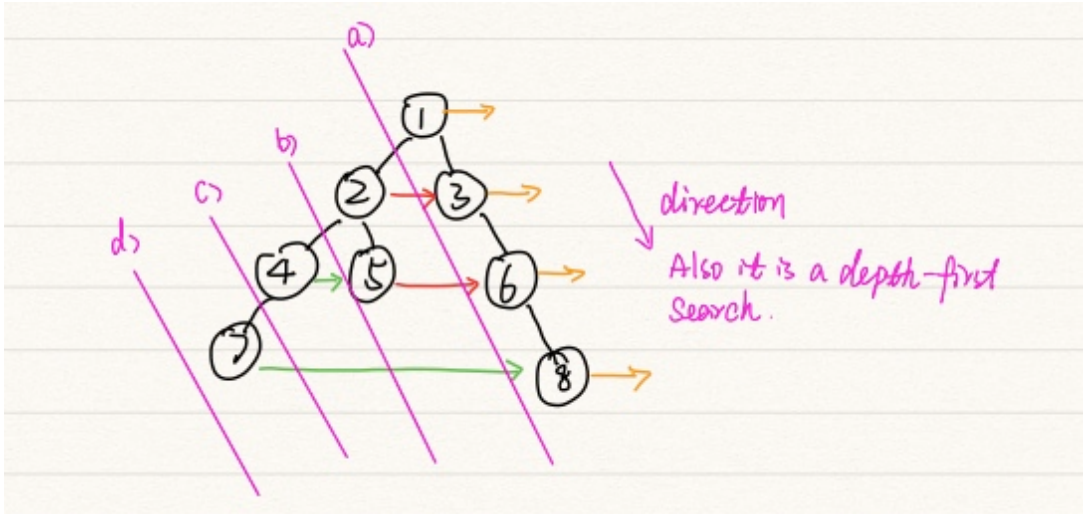
For example,
Given the following binary tree,

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

After calling your function, the tree should look like:

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

1. Thought line



2. Depth-first Search + Tree

```
/**
 * 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 {
private:
    void connect_fct(TreeLinkNode *node, TreeLinkNode* nodeNext){
        if (!node) return;
        node->next = nodeNext;

        while (nodeNext && !nodeNext->left && !nodeNext->right )
            nodeNext = nodeNext->next;
        TreeLinkNode* temp = (!nodeNext)?nullptr: ((!nodeNext->left)?nodeNext->right:nodeNext->left);

        if (node->right){
            connect_fct(node->right, temp);
            connect_fct(node->left, node->right);
        }
        else
            connect_fct(node->left, temp);
    }
public:
    void connect(TreeLinkNode *root) {
        connect_fct(root, nullptr);
        return;
    }
};
```