117 Populating Next Right Pointers in Each Node II

Problem

Write a function to connect all the adjacent nodes at the same level in a binary tree. The structure of the given Binary Tree node is like following. Initially, all the nextRight pointers point to garbage values. Your function should set these pointers to point next right for each node. You can use only constant extra space.

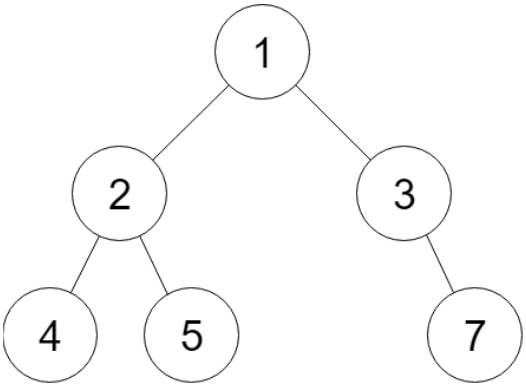

Figure A

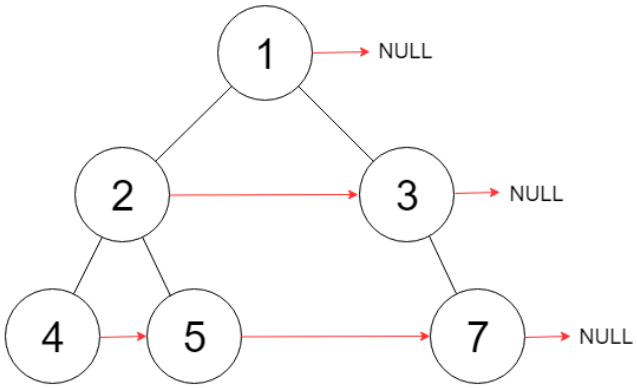

Figure B

Approach

Level Order Traversal (BFS)

- Use a Breadth-First Search (BFS) to traverse the tree level by level.
- For each level, connect each node to its next right node.
- Move to the next level and repeat.

Algorithm

- Start with the root node.
- Create a queue for BFS and add the root node to it.
- While the queue is not empty:
    - Get the size of the queue, which represents the number of nodes in the current level.
    - Initialize a variable `prev` as `NULL` to keep track of the previously visited node in this level.
    - For each node in this level:
        - Pop a node from the queue.
        - If `prev` is not `NULL`, set `prev.next` to the current node.
        - Update `prev` to be the current node.
        - If the current node has a left child, add it to the queue.
        - If the current node has a right child, add it to the queue.

- After this loop, `prev.next` will be `NULL`, which is correct as it's the last node in this level.

- Return the root node.

Implementation

```
class Solution:
    def connect(self, root: 'Node') -> 'Node':
        if not root:
            return None

        queue = [root]

        while queue:
            size = len(queue)
            prev = None

            for _ in range(size):
                node = queue.pop(0)

                if prev:
                    prev.next = node

                prev = node

                if node.left:
                    queue.append(node.left)
                if node.right:
                    queue.append(node.right)

            prev.next = None

        return root
```