Documentation for "Populating Next Right Pointers in Each Node II"

Problem Statement

Given a binary tree with the following structure:

```
struct Node {
  int val;
  Node *left;
  Node *right;
  Node *next;
}
```

The task is to populate each next pointer of the tree nodes 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.

Example 1

Input: root = [1, 2, 3, 4, 5, null, 7]

Output: [1, #, 2, 3, #, 4, 5, 7, #]

Explanation: Given the above binary tree (Figure A), the function should populate each next pointer to point to its next right node, just like in Figure B. The serialized output is in level order as connected by the next pointers, with # signifying the end of each level.

Example 2

Input: root = []

Output: []

Constraints

- The number of nodes in the tree is in the range [0, 6000].
- -100 <= Node.val <= 100

Follow-up

- You may only use constant extra space.
- The recursive approach is acceptable. Implicit stack space due to recursion does not count as extra space for this problem.

Solution

The solution involves using a level-order traversal approach to connect the next pointers of each node to its next right node. The traversal can be efficiently implemented using a queue to handle nodes level by level. This ensures that each node is connected to the appropriate next right node, or set to NULL if there is no next right node on that level.

The function operates by:

1. Checking if the root is None, returning None immediately if true.
2. Initializing a queue with the root node to facilitate level-order traversal.
3. Iterating through each level of the tree using a loop, processing each node at the current level.
4. Connecting the current node's next pointer to the next node in the queue if it is not the last node of the level.
5. Enqueueing the left and right children of the current node, if they exist.
6. Continuing this process until all levels are processed.

The result is a modified tree where each node's next pointer correctly points to its next right node.