LC 103. Binary Tree Zigzag Level Order Traversal

Question

Given a binary tree, return the *zigzag level order* traversal of its nodes' values. (ie, from left to right, then right to left for the next level and alternate between).

For example:
Given binary tree `[3,9,20,null,null,15,7]`,

```
    3
   / \
  9  20
    /  \
   15   7
```

return its zigzag level order traversal as:

```
[
  [3],
  [20,9],
  [15,7]
]
```

Solution

```python
# Definition for a binary tree node.
# class TreeNode:
#     def __init__(self, x):
#         self.val = x
#         self.left = None
#         self.right = None

class Solution:
    def zigzagLevelOrder(self, root: TreeNode) -> List[List[int]]:
        #Solution 2 - BFS
        if not root:
            return []
        queue = [root]
        res = []
        level = 0

        while queue:
            level += 1
            temp = []
            tempq = []
```

```
            for node in queue:
                temp.append(node.val)
                if node.left:
                    tempq.append(node.left)
                if node.right:
                    tempq.append(node.right)
            queue = tempq
            if level % 2 == 0:
                temp = temp[::-1]
            res.append(temp)

        return res


        #Solution
        if not root:
            return []
        stack = [root]
        ans = []
        reverse = False
        while stack:
            vals = [node.val for node in stack]
            if reverse:
                vals = vals[::-1]
            ans.append(vals)
            stack = [kid for node in stack for kid in (node.left, node.right) if
kid]
            reverse = not reverse
        return ans
```