Return binary 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,#,#,15,7},

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

return its zigzag level order traversal as:

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

Solutions:

```
class TreeNode:
    def __init__(self, x):
        self.val = x
        self.left = None
        self.right = None
class Solution:
    # @param root, a tree node
    # @return a list of lists of integers
    def zigzagLevelOrder(self, root):
        solution = []
        thisLevel =[]
```

```
        if root != None:
            thisLevel.append(root)
        leftToRight = True
        while len(thisLevel)>0:
            levelSolution = []
            nextLevel = []
            while len(thisLevel)>0:
                node = thisLevel.pop()
                levelSolution.append(node.val)
                if leftToRight:
                    if node.left != None:
                        nextLevel.append(node.left)
                    if node.right != None:
                        nextLevel.append(node.right)
                else:
                    if node.right != None:
                        nextLevel.append(node.right)
                    if node.left != None:
                        nextLevel.append(node.left)
            thisLevel = nextLevel
            solution.append(levelSolution)
            leftToRight = not leftToRight
        return solution
if __name__ == '__main__':
    BT, BT.left, BT.right, BT.right.left, BT.right.right = TreeNode(3), TreeNode(9), TreeNode(20), TreeNode(15), TreeNode(7)
    print ( Solution().zigzagLevelOrder(BT) )
```