103. Binary Tree Zigzag Level Order Traversal

Description (?tab=Description) | Submission (?tab=Submission) | Solutions (?tab=Solutions) | Add to List (?tab=List)

Total Accepted: **88118** Total Submissions: **269738** Difficulty: **Medium** Contributors: **Admin**

Notes

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]
]
```

Subscribe (/subscribe/) to see which companies asked this question.

Show Tags

Show Similar Problems

Have you met this question in a real interview? Yes No

Discuss (https://discuss.leetcode.com/category/111) | Pick One (/problems/random-one-question/) | Editorial Solution

Python

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

class Solution(object):
    def zigzagLevelOrder(self, root):
        """
        :type root: TreeNode
        :rtype: List[List[int]]
        """
        def bfs(root):
            d=[]#d is a list
            i=0
            thislevel=[root]
            while thislevel:
                d.append([]) #add a new list to d for a new level
                for node in thislevel:
                    d[i].append(node.val)
                if i%2==1:
                    d[i]=d[i][::-1]
                nextlevel=[]
                for node in thislevel:
                    if node.left:
                        nextlevel.append(node.left)
                    if node.right:
                        nextlevel.append(node.right)
                i=i+1
                thislevel=nextlevel
            return d
        if root:
            d=bfs(root)
            return d
        return []
```

Send Feedback (mailto:admin@leetcode.com?subject=Feedback)

Custom Testcase

Contribute Testcase

Run Code | Submit Solution

Run Code Status: Finished

Run Code Result:

Your input

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

Your answer

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

Expected answer

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

Show Diff

Runtime: 42 ms

Note: is Run Code inconsistent with Submit Solution? If you are using global variables or C/C++, check this (/faq/#different-output) out.

Notes