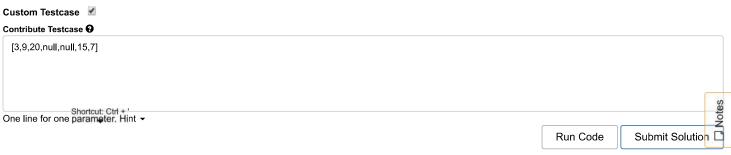
107. Binary Tree Level Order Traversal II

```
砂
                                                                      \odot
 Description (?tab=Description)
                              Submission (?tab=Submission)
                                                            Solutions (?tab=Solutions)
                                                                                      Add to List (?tab=List)
Total Accepted: 114361 Total Submissions: 299118 Difficulty: Easy
                                                                Contributors: Admin
Given a binary tree, return the bottom-up level order traversal of its nodes' values. (ie, from left to right, level by level from leaf to root).
For example:
Given binary tree [3,9,20,null,null,15,7],
     3
    / \
   9 20
     /
    15
return its bottom-up level order traversal as:
 [
   [15,7],
   [9,20],
   [3]
 1
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/115)

→ Pick One (/problems/random-one-question/)
                                                                                             Editorial Solution
 Python
                                    C
                                           </>
        # Definition for a binary tree node.
        # class TreeNode(object):
               def _
                    __init__(self, x):
self.val = x
self.left = None
    3
       #
    4
    5
        #
                    self.right = None
    6
        #
        class Solution(object):
    9
             def levelOrderBottom(self, root):
   10
   11
                  :type root: TreeNode
   12
                  :rtype: List[List[int]]
"""
   13
   14
                 def bfs(root):
   15
                      d={}
   16
                      i=0
   17
   18
                      thislevel=[root]
   19
                      while thislevel:
   20
                           d[i]=[]
   21
                           for node in thislevel:
                                d[i].append(node.val)
   22
   23
   24
                           i=i+1
                           nextlevel=[]
   25
                           for node in thislevel:
   26
                                if node.left:
   27
   28
                                    nextlevel.append(node.left)
   29
                                if node.right:
   30
                                    nextlevel.append(node.right)
   31
   32
                           thislevel=nextlevel
   33
                      return d
   34
   35
                 if root:
   36
                      d=bfs(root)

■ Send Feedback (mailto:admin@leetcode.com?subject=Feedback)
                      return d.values()[::-1]
```



Run Code Status: Finished



Frequently Asked Questions (/faq/) | Terms of Service (/tos/)

Privacy

Copyright © 2017 LeetCode

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