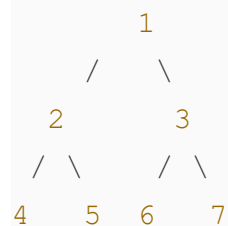


Vertical sum

Given a Binary Tree, find vertical sum of the nodes that are in same vertical line. Print all sums through different vertical lines starting from left-most vertical line to right-most vertical line.

Example 1:

Input:



Output:

Explanation:

The tree has 5 vertical lines

Vertical-Line-1 has only one node

4 => vertical sum is 4

Vertical-Line-2: has only one node

2=> vertical sum is 2

Vertical-Line-3: has three nodes:

1, 5, 6 => vertical sum is 1+5+6 = 12

Vertical-Line-4: has only one node 3

=> vertical sum is 3

Vertical-Line-5: has only one node 7

=> vertical sum is 7

Your Task:

You don't need to take input. Just complete the function **verticalSum()** that takes **root** node of the tree as parameter and returns an array containing the vertical sum of tree from left to right.

Expected Time Complexity: O(N).

Expected Auxiliary Space: O(N).

Constraints:

1<=Number of nodes<=1000

```
class Solution:
    #Complete the function below
    def verticalSum(self, root):
        #:param root: root of the given tree.
```

```
#code here
ans = {}
self.helper(root,ans,0)
# return
res = []
for key in sorted(ans.keys()):
    res.append(ans[key])
return res

def helper(self,root,ans,level):
    if root is None:
        return
    if level in ans:
        ans[level] = ans[level]+root.data
    else:
        ans[level] = root.data
    self.helper(root.left,ans,level-1)
    self.helper(root.right,ans,level+1)
```