Bottom View of Binary Tree

Given a binary tree, print the bottom view from left to right.
A node is included in bottom view if it can be seen when we look at the tree from bottom.

```
20
          /
        8      22
      /  \
    5      3      25
          /  \
        10    14
```

For the above tree, the bottom view is 5 10 3 14 25.
If there are **multiple** bottom-most nodes for a horizontal distance from root, then print the later one in level traversal. For example, in the below diagram, 3 and 4 are both the bottommost nodes at horizontal distance 0, we need to print 4.

```
20
          /
        8      22
      /  \    /
    5      3 4    25
           /   \
        10      14
```

For the above tree the output should be 5 10 4 14 25.

```python
def bottomView(root):

    # code here
    seen = {}
    least = [0]
    # minimum = least[0]
    helper(root, seen, 0,0)
    ans = []
    minimum = min(seen.keys())
    while True:
        if minimum in seen:
            ans.append(seen[minimum][0])
            minimum = minimum+1
        else:
            break
    return ans



def helper(root, seen, level,depth):
    if root is None:
        return

    helper(root.left, seen, level - 1,depth+1)
    helper(root.right, seen, level + 1,depth+1)
    if level in seen:
        if depth>=seen[level][1]:
            seen[level] = [root.data,depth]
    else:
        seen[level] = [root.data,depth]
```