BFS

Given the root of a binary tree, return the view you would see if you were standing on the right hand side of it ordered from top to bottom (assume a node blocks all nodes to the left of it so you can only see the right-most node of a level).

Q = (1, 0)

(2, 1) (5, 1)

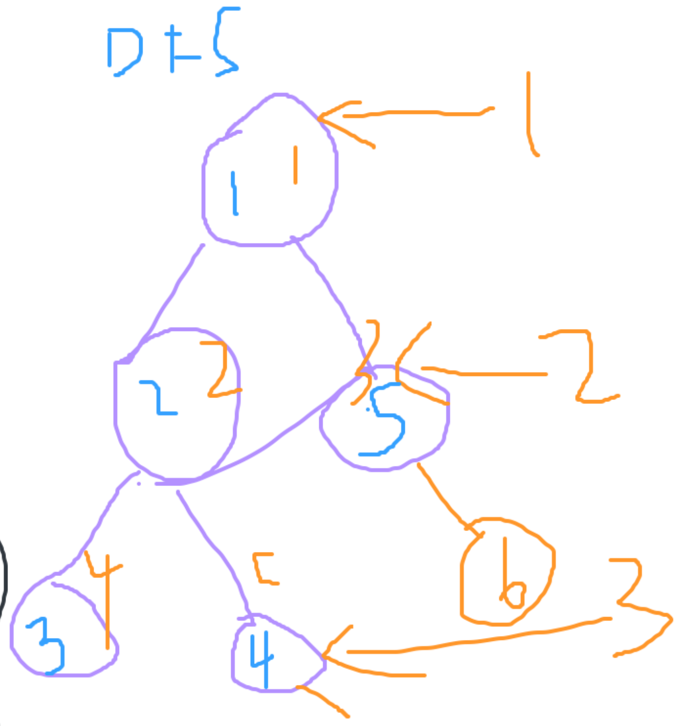
(3, 2) (4, 2)

(6, 2)

```
  # breadth first search
  # for each level, go L to Right
  # append the result accordingly
  # []
  # [ 1]
  #[1, 2] see 5, update so now [1,5]
  # [ 1,5,3] see 4 update [1,5,4]
  # if we go to a new level, append otherwise replace

def right_side_view(root_node):
    if not root_node:
       return []
    queue = deque([(root, 0)])
    visited = seen()
   result = []
   previous_level = -1
    while queue:
      # current = deque.pop_left
     # current_node = current[0]
    # current_level = current[1]
      current_node, current_level = deque.popleft()
     if current_level != previous_level:
           result.append(current_node)
     else:
          result[-1] = current_node
    if current_node.left:
          queue.append((current_node.left, current_level+ 1)
    if current_node.right:
           queue.append((current_node.right, current_level + 1))
   previous_level = current_level
   return result
```

*


DFS
1
2
5
3
4
6
1
2
3