12323 - LAB 04

Instructions

1. Access the auto-grader at https://c200.luddy.indiana.edu
2. Please write the code for the problems in python language
3. The code should be readable with variables named meaningfully
4. Plagiarism is unacceptable and we have ways to find it, so do not do it
5. Don't change the function signature (name of the function and number and types of arguments) provided in this file.
6. Once you pass all the tests on the auto grader, show your work to the teaching assistant

Problem

Question

Jeff is facing a binary tree, and he's contemplating what the tree would look like from his perspective if he were standing on the right side. In this scenario, the binary tree may or may not have child nodes. To assist Jeff in visualizing the right view of the tree, a function named rightView(node) needs to be implemented. This function takes root of the binary tree as its argument and outputs a list of nodes that would be visible to Jeff if he were standing on the right side. The goal is to provide a clear, ordered representation of the nodes, ensuring that the list reflects the top-to-bottom visibility of the tree from Jeff's vantage point.

Test cases

Input: grid = [9, 8, 7, None, None, 7, 8] // you would be given an object of class treenode
Output:[9,7,8]
Explanation: Refer Figure-1 on page-2

Input:grid =[1, 2, None, 4, None, 5, None, 6] //you would be given an object of class TreeNode
Output: [1,2,4,5,6]
Explanation: Refer Figure-2 on page-2

Function signature

```
# While writing the code always include the below class
class TreeNode:
    def __init__(self, val=0, left=None, right=None):
        self.val = val
        self.left = left
        self.right = right
def rightView(root:TreeNode)-> list[int]:
    pass
```

Constraints

1.) Solve the question in O(n) where n is the number of nodes in the tree.

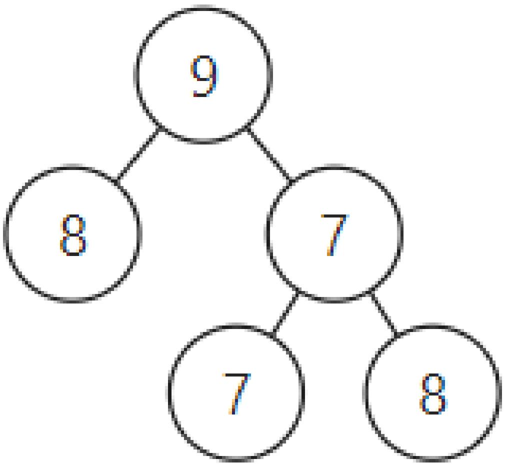

Figure 1: Test case 1

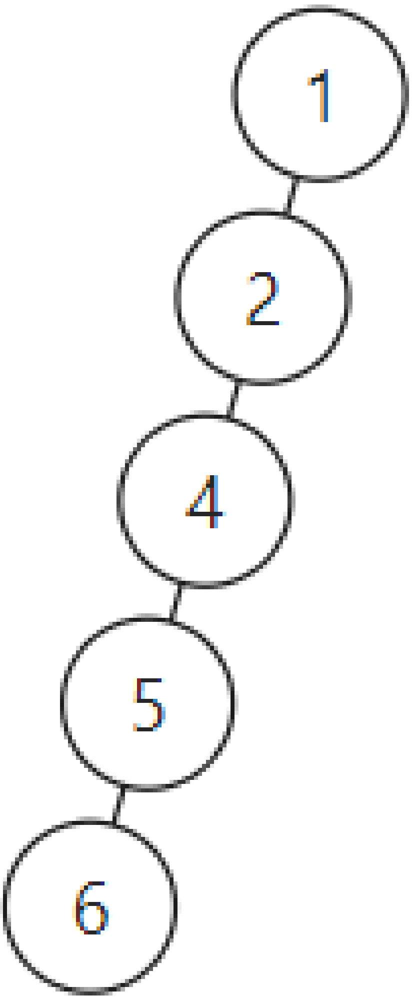

Figure 2: Test case 2