Problem 662: Maximum Width of Binary Tree

Problem Information

Difficulty: Medium
Acceptance Rate: 44.91%
Paid Only: No
Tags: Tree, Depth-First Search, Breadth-First Search, Binary Tree

Problem Description

Given the `root` of a binary tree, return _the**maximum width** of the given tree_.

The **maximum width** of a tree is the maximum **width** among all levels.

The **width** of one level is defined as the length between the end-nodes (the leftmost and rightmost non-null nodes), where the null nodes between the end- nodes that would be present in a complete binary tree extending down to that level are also counted into the length calculation.

It is **guaranteed** that the answer will in the range of a **32-bit** signed integer.

Example 1:

Input: root = [1,3,2,5,3,null,9] **Output:** 4 **Explanation:** The maximum width exists in the third level with length 4 (5,3,null,9).

Example 2:

Input: root = [1,3,2,5,null,null,9,6,null,7] **Output:** 7 **Explanation:** The maximum width exists in the fourth level with length 7 (6,null,null,null,null,null,7).

Example 3:

Input: root = [1,3,2,5] **Output:** 2 **Explanation:** The maximum width exists in the second level with length 2 (3,2).

Constraints:

* The number of nodes in the tree is in the range `[1, 3000]`. * `-100 <= Node.val <= 100`

Code Snippets

C++:

```
/**
* Definition for a binary tree node.
* struct TreeNode {
* int val;
* TreeNode *left;
* TreeNode *right;
* TreeNode() : val(0), left(nullptr), right(nullptr) {}
* TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}
* TreeNode(int x, TreeNode *left, TreeNode *right) : val(x), left(left),
right(right) {}
* };
*/
class Solution {
public:
int widthOfBinaryTree(TreeNode* root) {

}
};
```

Java:

```
/**
* Definition for a binary tree node.
* public class TreeNode {
* int val;
* TreeNode left;
* TreeNode right;
```

```
* TreeNode() {}
* TreeNode(int val) { this.val = val; }
* TreeNode(int val, TreeNode left, TreeNode right) {
* this.val = val;
* this.left = left;
* this.right = right;
* }
* }
*/
class Solution {
public int widthOfBinaryTree(TreeNode root) {

}
}
```

Python3:

```
# Definition for a binary tree node.
# class TreeNode:
# def __init__(self, val=0, left=None, right=None):
# self.val = val
# self.left = left
# self.right = right
class Solution:
def widthOfBinaryTree(self, root: Optional[TreeNode]) -> int:
```