662. Maximum Width of Binary Tree

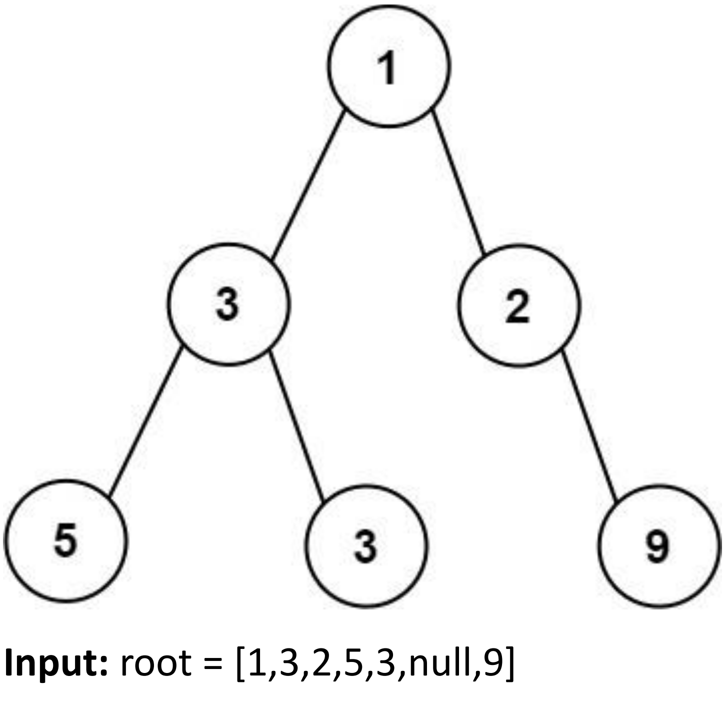
Given the root of a binary tree, return *the **maximum width** of the given tree*.

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:

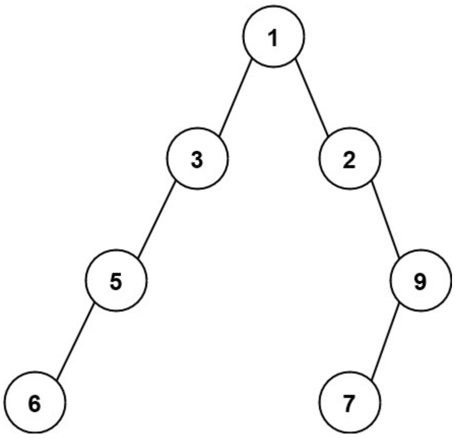


Input: root = [1,3,2,5,3,null,9]

Output: 4

Explanation: The maximum width exists in the third level with length 4 (5,3,null,9).

Example 2:

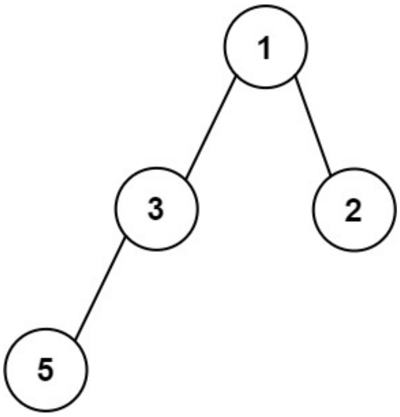


Input: root = [1,3,2,5,null,null,9,6,null,7]

Output: 7

Explanation: The maximum width exists in the fourth level with length 7 (6,null,null,null,null,null,7).

Example 3:



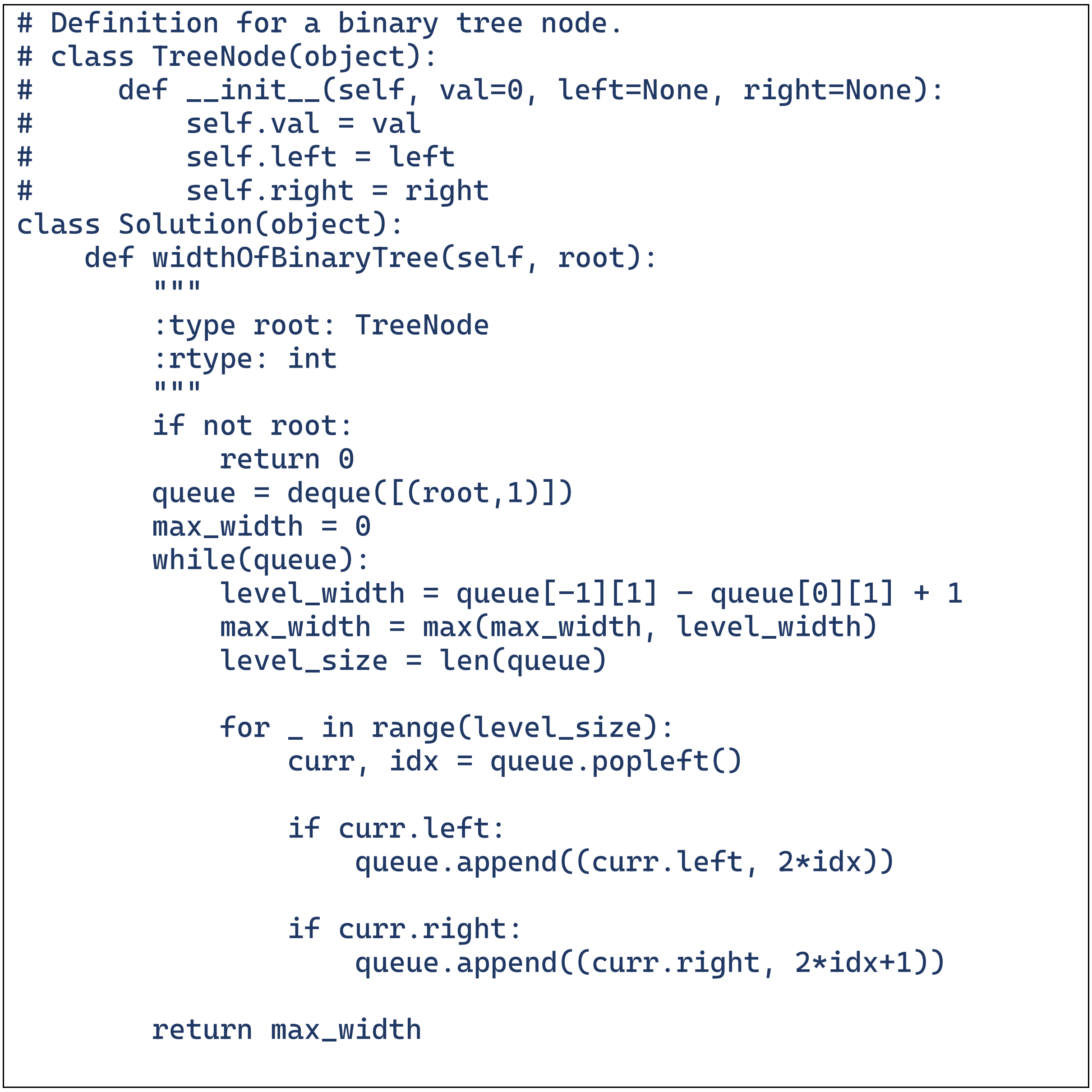
Input: root = [1,3,2,5]

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



```
# Definition for a binary tree node.
# class TreeNode(object):
#     def __init__(self, val=0, left=None, right=None):
#         self.val = val
#         self.left = left
#         self.right = right
class Solution(object):
    def widthOfBinaryTree(self, root):
        """
        :type root: TreeNode
        :rtype: int
        """
        if not root:
            return 0
        queue = deque([(root,1)])
        max_width = 0
        while(queue):
            level_width = queue[-1][1] - queue[0][1] + 1
            max_width = max(max_width, level_width)
            level_size = len(queue)

            for _ in range(level_size):
                curr, idx = queue.popleft()

                if curr.left:
                    queue.append((curr.left, 2*idx))

                if curr.right:
                    queue.append((curr.right, 2*idx+1))

        return max_width
```