

```

/**
 * Definition for a binary tree node.
 * public class TreeNode {
 *     int val;
 *     TreeNode left;
 *     TreeNode right;
 *     TreeNode(int x) { val = x; }
 * }
 */
class Solution {
    public int widthOfBinaryTree(TreeNode root) {
        Queue<TreeNode> q = new LinkedList<>();
        Deque<Integer> dq = new ArrayDeque<>();
        q.offer(root);
        dq.offer(1);
        int cur_max = 0;
        while (!q.isEmpty()) {
            int size = q.size();
            cur_max = Math.max(cur_max, dq.peekLast() - dq.peekFirst() + 1);
            for (int i = 0; i < size; i++) {
                TreeNode node = q.poll();
                int val = dq.pollFirst();
                if (node.left != null) {
                    q.offer(node.left);
                    dq.offerLast(val * 2);
                }
                if (node.right != null) {
                    q.offer(node.right);
                    dq.offerLast(val * 2 + 1);
                }
            }
        }
        return cur_max;
    }
}

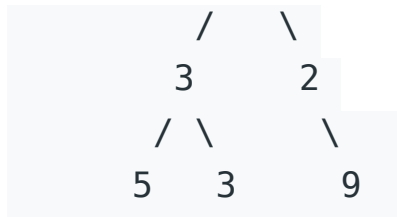
```

Given a binary tree, write a function to get the maximum width of the given tree. The width of a tree is the maximum width among all levels. The binary tree has the same structure as a **full binary tree**, but some nodes are null.

The width of one level is defined as the length between the end-nodes (the leftmost and right most non-null nodes in the level, where the `null` nodes between the end-nodes are also counted into the length calculation).

Example 1:

Input:



Output: 4

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