1. **Maximum Width of Tree**

class Solution {

    public int widthOfBinaryTree(TreeNode root) {

        if (root == null) return 0;

        int maxWidth = 0;

        Queue<Pair<TreeNode, Integer>> q = new LinkedList<>();

        q.offer(new Pair<>(root, 0));

        while (!q.isEmpty()) {

            int size = q.size();

            int minIndex = q.peek().getValue();

            int first = 0, last = 0;

            for (int i = 0; i < size; i++) {

                Pair<TreeNode, Integer> current = q.poll();

                TreeNode node = current.getKey();

                int index = current.getValue() - minIndex;

                if (i == 0) first = index;

                if (i == size - 1) last = index;

                if (node.left != null) {

                    q.offer(new Pair<>(node.left, 2 \* index + 1));

                }

                if (node.right != null) {

                    q.offer(new Pair<>(node.right, 2 \* index + 2));

                }

            }

            maxWidth = Math.max(maxWidth, last - first + 1);

        }

        return maxWidth;

    }

}