```
/*

    Time complexity: O(N)
    Space complexity: O(H)

    where N is the number of nodes in the input tree
    and H is the height of the input tree

*/


public class Solution {

        public static Pair<Integer, Integer> getMinAndMax(BinaryTreeNode<Integer> root) {

            if (root == null) {
                return new Pair<>(Integer.MAX_VALUE, Integer.MIN_VALUE);
            }

            Pair<Integer, Integer> leftPair = getMinAndMax(root.left);
            Pair<Integer, Integer> rightPair = getMinAndMax(root.right);

            int minimum = Math.min(root.data, Math.min(leftPair.minimum, rightPair.minimum));
            int maximum = Math.max(root.data, Math.max(leftPair.maximum, rightPair.maximum));

            return new Pair<>(minimum, maximum);

        }

}
```