

```

// Function to return the diameter of a Binary Tree.
// Node{int data, Node left, Node right}
int diameter(Node root) {
    if (root == null) {
        return 0;
    }
    // height of left and right subtree.
    int leftHeight = heightOfBinaryTree(root.left);
    int rightHeight = heightOfBinaryTree(root.right);
    // diameter of left and right subtree.
    int leftDiameter = diameter(root.left);
    int rightDiameter = diameter(root.right);
    // diameter of the subtree with root.
    int maxDiameter1 = leftHeight + rightHeight + 1;
    // diameter of the subtree without root.
    int maxDiameter2 = Math.max(leftDiameter, rightDiameter);

    return Math.max(maxDiameter1, maxDiameter2);
}

// Function to find the height of a binary tree.
// Node{int data, Node left, Node right}
int heightOfBinaryTree(Node node) {
    if (node == null) {
        return 0;
    }
    return Math.max(heightOfBinaryTree(node.left), heightOfBinaryTree(node.right)) + 1;
}

```