Problem 3: Find the Diameter of a Binary Tree. **Diameter** is the length of the longest path between any 2 nodes in the tree and this path may or may not pass from the root.

```
class Node:
    def __init__(self, data):
        self.data = data
        self.left = None
        self.right = None


def height(node):
    if node is None:
        return 0
    return max(height(node.left), height(node.right)) + 1


def diameter(node):
    if node is None:
        return 0

    left_height = height(node.left)
    right_height = height(node.right)

    left_diameter = diameter(node.left)
    right_diameter = diameter(node.right)

    return max(left_height + right_height + 1, max(left_diameter, right_diameter))

root = Node(1)
root.left = Node(2)
root.right = Node(3)
root.left.left = Node(4)
root.left.right = Node(5)
```

root.right.left = Node(6)

root.right.right = Node(7)

print("Diameter of the binary tree is:", diameter(root))