

### Problem 5: Size of the largest BST in a Binary Tree

```
class Node:
```

```
    def __init__(self, value):
```

```
        self.data = value
```

```
        self.left = None
```

```
        self.right = None
```

```
def largestBSTSize(root):
```

```
    def isBST(node, min_value, max_value):
```

```
        if node is None:
```

```
            return True
```

```
        if node.data < min_value or node.data > max_value:
```

```
            return False
```

```
        return (
```

```
            isBST(node.left, min_value, node.data - 1)
```

```
            and isBST(node.right, node.data + 1, max_value)
```

```
        )
```

```
def countNodes(node):
```

```
    if node is None:
```

```
        return 0
```

```
    return 1 + countNodes(node.left) + countNodes(node.right)
```

```
def largestBSTSizeUtil(node):
```

```
    if isBST(node, float("-inf"), float("inf")):
```

```
        return countNodes(node)
```

```
    return max(
```

```
        largestBSTSizeUtil(node.left),  
        largestBSTSizeUtil(node.right)  
    )
```

```
return largestBSTSizeUtil(root)
```

```
root = Node(6)
```

```
root.left = Node(4)
```

```
root.right = Node(7)
```

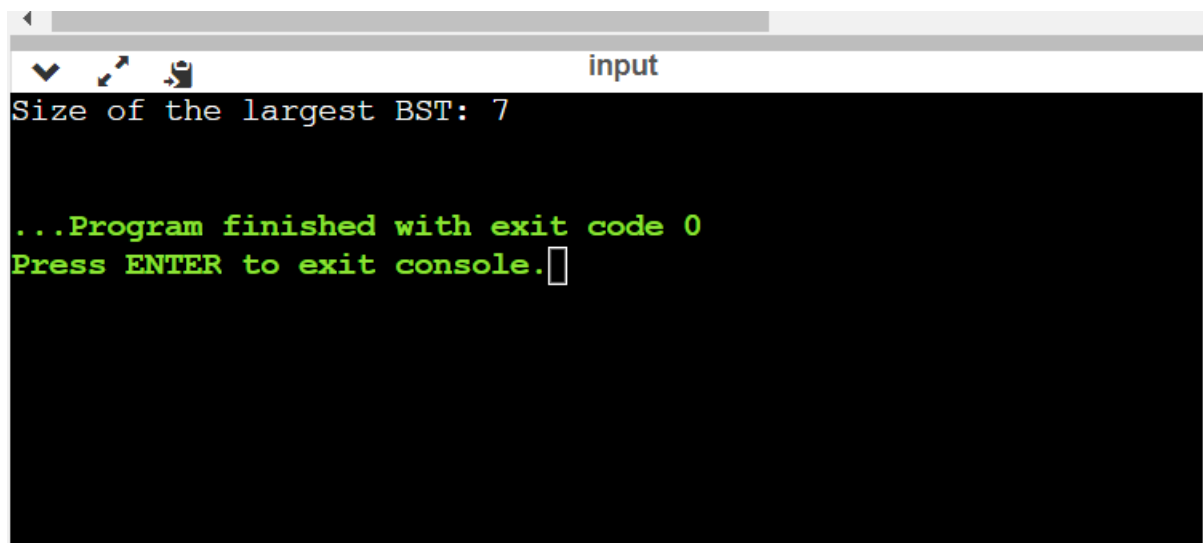
```
root.left.left = Node(3)
```

```
root.left.right = Node(5)
```

```
root.right.right = Node(9)
```

```
root.right.right.left = Node(8)
```

```
print("Size of the largest BST:", largestBSTSize(root))
```



```
Size of the largest BST: 7  
  
...Program finished with exit code 0  
Press ENTER to exit console.□
```