Problem Challenge 1

We'll cover the following

- Tree Diameter (medium)
- Try it yourself

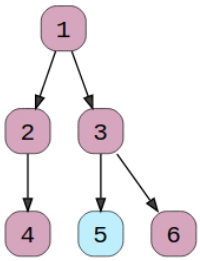
Tree Diameter (medium)

Given a binary tree, find the length of its diameter. The diameter of a tree is the number of nodes on the **longest path between any two leaf nodes**. The diameter of a tree may or may not pass through the root.

Note: You can always assume that there are at least two leaf nodes in the given tree.

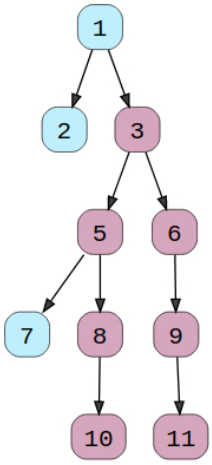
Example 1:

Output: 5
Explanation: The diameter of the tree is: [4, 2, 1, 3, 6]



Example 2:

Output: 7
Explanation: The diameter of the tree is: [10, 8, 5, 3, 6, 9, 11]



Try it yourself

Try solving this question here:

Java | Python3 | JS | C++

```
class TreeNode:
  def __init__(self, val, left=None, right=None):
    self.val = val
    self.left = left
    self.right = right


class TreeDiameter:

```

```
  def __init__(self):
    self.treeDiameter = 0

  def find_diameter(self, root):
    # TODO: Write your code here
    return -1

def main():
  treeDiameter = TreeDiameter()
  root = TreeNode(1)
  root.left = TreeNode(2)
  root.right = TreeNode(3)
  root.left.left = TreeNode(4)
  root.right.left = TreeNode(5)
  root.right.right = TreeNode(6)
  print("Tree Diameter: " + str(treeDiameter.find_diameter(root)))
  root.left.left = None
  root.right.left.left = TreeNode(7)
  root.right.left.right = TreeNode(8)
  root.right.right.left = TreeNode(9)
  root.right.left.right.left = TreeNode(10)
  root.right.right.left.left = TreeNode(11)
  print("Tree Diameter: " + str(treeDiameter.find_diameter(root)))


main()
```

Run Save Reset

Back

Count Paths for a Sum (medium)

Next

Solution Review: Problem Challenge 1

Completed