Problem Challenge 2

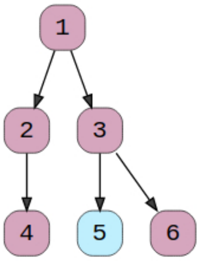
Path with Maximum Sum (hard)

Find the path with the maximum sum in a given binary tree. Write a function that returns the maximum sum.

A path can be defined as a **sequence of nodes between any two nodes** and doesn't necessarily pass through the root. The path must contain at least one node.

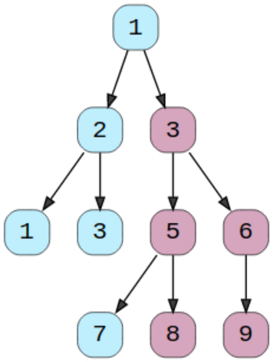
Example 1:

Output: 16
Explanation: The path with maximum sum is: [4, 2, 1, 3, 6]



Example 2:

Output: 31
Explanation: The path with maximum sum is: [8, 5, 3, 6, 9]



Try it yourself

Try solving this question here:

Java | Python3 | JS | C++

```python
import math


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


def find_maximum_path_sum(root):
    # TODO: Write your code here
    return -1


def main():
    root = TreeNode(1)
    root.left = TreeNode(2)
    root.right = TreeNode(3)

    print("Maximum Path Sum: " + str(find_maximum_path_sum(root)))
    root.left.left = TreeNode(1)
    root.left.right = TreeNode(3)
    root.right.left = TreeNode(5)
    root.right.right = TreeNode(6)
    root.right.left.left = TreeNode(7)
    root.right.left.right = TreeNode(8)
    root.right.right.left = TreeNode(9)
    print("Maximum Path Sum: " + str(find_maximum_path_sum(root)))

    root = TreeNode(-1)
    root.left = TreeNode(-3)
    print("Maximum Path Sum: " + str(find_maximum_path_sum(root)))


main()
```

Run Save Reset

← Back
Solution Review: Problem Challenge 1

Next →
Solution Review: Problem Challenge 2

☑ Completed

Report an Issue Ask a Question