

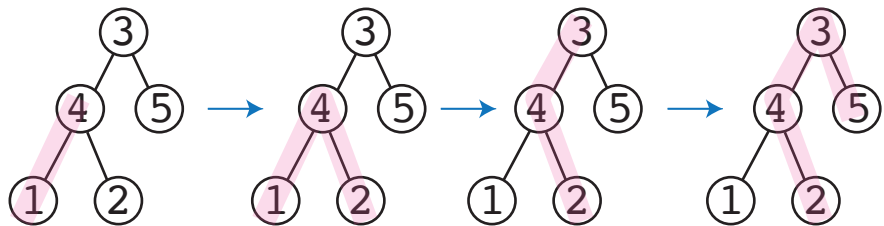
124. Binary Tree Maximum Path Sum

Hard

A path in a binary tree is a sequence of nodes where each pair of adjacent nodes in the sequence has an edge connecting them. A node can only appear in the sequence at most once. Note that the path does not need to pass through the root.

The path sum of a path is the sum of the node's values in the path.

Given the root of a binary tree, return the maximum path sum of any non-empty path.



Definition for a binary tree node.

```
class TreeNode:
```

```
    def __init__(self, val=0, left=None, right=None):
        self.val = val
        self.left = left
        self.right = right
```

```
class Solution:
```

```
    def maxPathSum(self, root: Optional[TreeNode]) -> int:
        result = [root.val]
```

```
    def dfs(node):
```

```
        if not node:
```

```
            return 0
```

```
        leftMax = dfs(node.left)
```

```
        rightMax = dfs(node.right)
```

```
        # compute max path sum with split
```

```
        leftMax = max(leftMax, 0)
```

```
        rightMax = max(rightMax, 0)
```

```
        result[0] = max(result[0], node.val + leftMax + rightMax)
```

```
        return node.val + max(leftMax, rightMax)
```

```
    dfs(root)
```

```
    return result[0]
```

get the max() of

-> current result[0]

-> node.val + leftMax + rightMax

