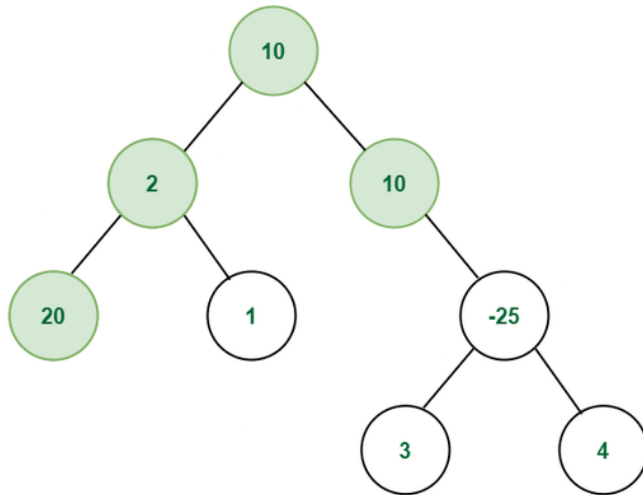

124 Maximum Path Sum in a Binary Tree

Given a binary tree, find the maximum path sum. The path may start and end at any node in the tree.



Approach

For each node there can be four ways that the max path goes through the node:

- Node only

- Max path through Left Child + Node

- Max path through Right Child + Node

- Max path through Left Child + Node + Max path through Right Child

The idea is to keep track of four paths and pick up the max one in the end. An important thing to note is, that the root of every subtree needs to return the maximum path sum such that at most one child of the root is involved (subroot + left, subroot+right, subroot). This is needed for the parent function call. If we also returned subroot + left + child, then the path would be going through the subroot twice.

Algorithm

FindMaxPath

```
max_val = int.MIN
```

```
findMax(root)
```

```
if root is Null return Null
```

```
l = findMax(root.left)
r = findMax(root.right)
max_single = max(max(l,r) + root.val, root.val)
max_top = max(max_single, l+r +root.val)
max_val = max(max_top, max_val)
return max_single
```