AlgoExpert Quad Layout Go 12px Sublime Monokai 00:00:00

Prompt Scratchpad Our Solution(s) Video Explanation Run Code

Solution 1

39 }

```
_{\rm 1} // Copyright @ 2020 AlgoExpert, LLC. All rights reserved.
   package main
 5 type BinaryTree struct {
    Value
               int
    Left, Right *BinaryTree
8 }
10 // O(n) time | O(log(n)) space
11 func MaxPathSum(tree *BinaryTree) int {
     _, maxSum := findMaxSum(tree)
13
    return maxSum
14 }
15
16 func findMaxSum(tree *BinaryTree) (int, int) {
     if tree == nil {
17
18
       return 0, 0
19
20
      leftMaxSumAsBranch, leftMaxPathSum := findMaxSum(tree.Left)
21
      rightMaxSumAsBranch, rightMaxPathSum := findMaxSum(tree.Right)
22
      \verb|maxChildSumAsBranch| := \verb|max(leftMaxSumAsBranch|, rightMaxSumAsBranch|)|
23
24
     value := tree.Value
25
      maxSumAsBranch := max(maxChildSumAsBranch+value, value)
26
     maxSumAsRootNode := max(leftMaxSumAsBranch+value+rightMaxSumAsBranch, maxSumAsBranch)
27
      maxPathSum := max(leftMaxPathSum, rightMaxPathSum, maxSumAsRootNode)
28
29
      return maxSumAsBranch, maxPathSum
30 }
31
32 func max(first int, vals \dotsint) int {
     for _, val := range vals {
33
34
       if val > first {
35
         first = val
36
37
38
     return first
```