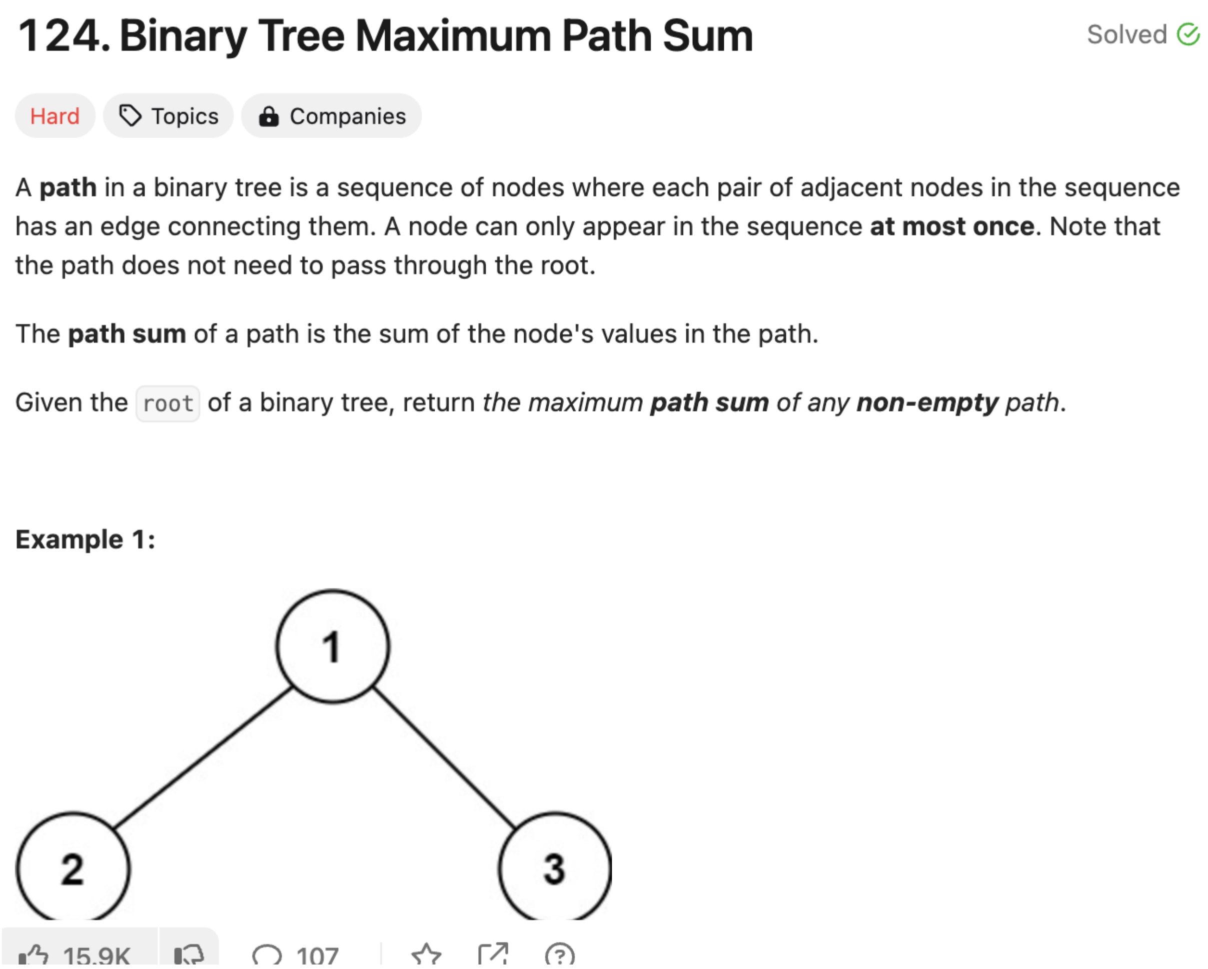

124. Binary Tree Maximum Path Sum

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*.

Example 1:

Code

```
/**
* Definition for a binary tree node.
* public class TreeNode {
*     public var val: Int
*     public var left: TreeNode?
*     public var right: TreeNode?
*     public init() { self.val = 0; self.left = nil; self.right = nil; }
*     public init(_ val: Int) { self.val = val; self.left = nil; self.right =
nil; }
*     public init(_ val: Int, _ left: TreeNode?, _ right: TreeNode?) {
*         self.val = val
*         self.left = left
*         self.right = right
*     }
* }
*/
```

```
class Solution {
   func maxPathSum(_ root: TreeNode?) -> Int {
       var maxsum = root!.val

       func dfs(_ root: TreeNode?) -> Int {
           guard let root = root else {
               return 0
           }

           var left = dfs(root.left)
           var right = dfs(root.right)
           left = max(0,left)
           right = max(0,right)

           maxsum = max(maxsum,(left + right + root.val))

           return root.val + max(left,right)
       }

       let _ = dfs(root)
       return maxsum
   }
}
```