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

 Prompt
 Scratchpad
 Our Solution(s)
 Video Explanation

Run Code

```
Solution 3
  Solution 1 Solution 2
  _{\rm 1} \, // Copyright @ 2020 AlgoExpert, LLC. All rights reserved.
     class Program {
          // O(n^2) time | O(1) space
          func \ numberOfBinaryTreeTopologies(\_ n: Int) \ -> \ Int \ \{
             var cache = [0: 1]
              return numberOfBinaryTreeTopologiesHelper(n, &cache)
          func numberOfBinaryTreeTopologiesHelper(_ n: Int, _ cache: inout [Int: Int]) -> Int {
10
11
              if let cachedValue = cache[n] {
12
                  return cachedValue
13
14
              var numberOfTopologies = 0
15
16
17
              for leftTreeSize in 0 ... < n \{
18
                  let rightTreeSize = n - 1 - leftTreeSize
19
20
21
                  let leftNumberOfTopologies = numberOfBinaryTreeTopologiesHelper(leftTreeSize, &cache)
let rightNumberOfTopologies = numberOfBinaryTreeTopologiesHelper(rightTreeSize, &cache)
22
                   numberOfTopologies += leftNumberOfTopologies * rightNumberOfTopologies
23
24
25
              cache[n] = numberOfTopologies
26
              return numberOfTopologies
27
28 }
29
```