

Solution 1Solution 2Solution 3

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 class Program {
4     // O(n^2) time | O(1) space
5     func numberOfBinaryTreeTopologies(_ n: Int) -> Int {
6         var cache = [1]
7
8         for currentMax in stride(from: 1, through: n, by: 1) {
9             var numberOfTopologies = 0
10
11             for leftTreeSize in stride(from: 0, to: currentMax, by: 1) {
12                 let rightTreeSize = currentMax - 1 - leftTreeSize
13
14                 let leftNumberOfTopologies = cache[leftTreeSize]
15                 let rightNumberOfTopologies = cache[rightTreeSize]
16                 numberOfTopologies += leftNumberOfTopologies * rightNumberOfTopologies
17             }
18
19             cache.append(numberOfTopologies)
20         }
21
22         return cache[n]
23     }
24 }
25
```

