Our Solution(s)   Run Code

Solution 1   Solution 2   Solution 3

```swift
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

class Program {
    // O(n^2) time | O(1) space
    func numberOfBinaryTreeTopologies(_ n: Int) -> Int {
        var cache = [1]

        for currentMax in stride(from: 1, through: n, by: 1) {
            var numberOfTopologies = 0

            for leftTreeSize in stride(from: 0, to: currentMax, by: 1
                let rightTreeSize = currentMax - 1 - leftTreeSize

                let leftNumberOfTopologies = cache[leftTreeSize]
                let rightNumberOfTopologies = cache[rightTreeSize]
                numberOfTopologies += leftNumberOfTopologies * rightN
            }

            cache.append(numberOfTopologies)
        }

        return cache[n]
    }
}
```

Your Solutions   Run Code

Solution 1   Solution 2   Solution 3

```swift
class Program {
    func numberOfBinaryTreeTopologies(_ n: Int) -> Int {
        // Write your code here.
        return -1
    }
}
```

Our Tests

Custom Output   Submit Code

Run or submit code when you're ready.