Our Solution(s)   Run Code

Solution 1   Solution 2   Solution 3

```python
# Copyright © 2020 AlgoExpert, LLC. All rights reserved.

# O(n^2) time | O(n) space
def numberOfBinaryTreeTopologies(n):
    cache = [1]
    for m in range(1, n + 1):
        numberOfTrees = 0
        for leftTreeSize in range(m):
            rightTreeSize = m - 1 - leftTreeSize
            numberOfLeftTrees = cache[leftTreeSize]
            numberOfRightTrees = cache[rightTreeSize]
            numberOfTrees += numberOfLeftTrees * numberOfRightTrees
        cache.append(numberOfTrees)
    return cache[n]
```

Your Solutions   Run Code

Solution 1   Solution 2   Solution 3

```python
def numberOfBinaryTreeTopologies(n):
    # Write your code here.
    pass
```

Our Tests

Custom Output   Submit Code

```python
def numberOfBinaryTreeTopologies(n):
    # Write your code here.
    pass
```

Run or submit code when you're ready.