```
1  // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3  // O(n^2) time | O(n) space
4  function numberOfBinaryTreeTopologies(n, cache = {0: 1}) {
5    if (n in cache) return cache[n];
6    let numberOfTrees = 0;
7    for (let leftTreeSize = 0; leftTreeSize < n; leftTreeSize++) {
8      const rightTreeSize = n - 1 - leftTreeSize;
9      const numberOfLeftTrees = numberOfBinaryTreeTopologies(leftTreeSize, cache);
10     const numberOfRightTrees = numberOfBinaryTreeTopologies(rightTreeSize, cache);
11     numberOfTrees += numberOfLeftTrees * numberOfRightTrees;
12   }
13   cache[n] = numberOfTrees;
14   return numberOfTrees;
15 }
16
17 exports.numberOfBinaryTreeTopologies = numberOfBinaryTreeTopologies;
18
```