

Solution 1Solution 2Solution 3

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 class Program {
4     // Upper Bound: O((n*(2n!))/(n!(n+1)!)) time | O(n) space
5     public static int numberOfBinaryTreeTopologies(int n) {
6         if (n == 0) {
7             return 1;
8         }
9         int numberOfTrees = 0;
10        for (int leftTreeSize = 0; leftTreeSize < n; leftTreeSize++) {
11            int rightTreeSize = n - 1 - leftTreeSize;
12            int numberOfLeftTrees = numberOfBinaryTreeTopologies(leftTreeSize);
13            int numberOfRightTrees = numberOfBinaryTreeTopologies(rightTreeSize);
14            numberOfTrees += numberOfLeftTrees * numberOfRightTrees;
15        }
16        return numberOfTrees;
17    }
18 }
19
```

