

Our Solution(s)

Run Code

Your Solutions

Run Code

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(leftTreeSiz
13            int numberOfRightTrees = numberOfBinaryTreeTopologies(rightTreeS
14            numberOfTrees += numberOfLeftTrees * numberOfRightTrees;
15        }
16        return numberOfTrees;
17    }
18 }
19
```

Solution 1Solution 2Solution 3

```
1 class Program {
2     public static int numberOfBinaryTreeTopologies(int n) {
3         // Write your code here.
4         return -1;
5     }
6 }
7
```

Our Tests

Custom Output

Submit Code

```
1 class Program {
2     public static int numberOfBinaryTreeTopologies(int n) {
3         // Write your code here.
4         return -1;
5     }
6 }
7
```

```
10 # Write event to test program under test using test program 10
11 }
12
13 # Test
14 # Write event to test program under test using test program 10
15 }
16
17 # Test
18 # Write event to test program under test using test program 10
19 }
20
21 # Test
```

Run or submit code when you're ready.