

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1Solution 2Solution 3

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 public 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 public class Program {
2     public static int NumberOfBinaryTreeTopologies(int n) {
3         // Write your code here.
4         return -1;
5     }
6 }
7
```

```
10  # Write down the first 10 terms of the sequence in a list
11  l = []
12
13  # Loop
14  for i in range(10):
15      # Write down the first 10 terms of the sequence in a list
16      l.append(1/i)
17
18  # Print the list
19  print(l)
```

Run or submit code when you're ready.