

PromptScratchpadOur Solution(s)Video Explanation

Run Code

Solution 1Solution 2Solution 3

```
1  # Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3  # Upper Bound: O((n*(2n!))/(n!(n+1)!)) time | O(n) space
4  def numberOfBinaryTreeTopologies(n):
5      if n == 0:
6          return 1
7      numberOfTrees = 0
8      for leftTreeSize in range(n):
9          rightTreeSize = n - 1 - leftTreeSize
10         numberOfLeftTrees = numberOfBinaryTreeTopologies(leftTreeSize)
11         numberOfRightTrees = numberOfBinaryTreeTopologies(rightTreeSize)
12         numberOfTrees += numberOfLeftTrees * numberOfRightTrees
13     return numberOfTrees
14
```

