

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1Solution 2Solution 3

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

Solution 1Solution 2Solution 3

```
1 package main
2
3 func NumberOfBinaryTreeTopologies(n int) int {
4     // Write your code here.
5     return -1
6 }
7
```

Our Tests

Custom Output

Submit Code

```
11 Run in JupyterLab Notebook (Python 3)
12 importlib = 2
13 importlib = NotebookView(importlib)
14 importlib.importlib, importlib, importlib
15 2
16
17 Run in JupyterLab Notebook (Python 3)
18 importlib = 2
19 importlib = NotebookView(importlib)
20 importlib.importlib, importlib, importlib
21 2
22
23 Run in JupyterLab Notebook (Python 3)
24 importlib = 2
```

Run or submit code when you're ready.