

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

Solution 2

Solution 3

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 import java.util.*;
4
5 class Program {
6     // O(n^2) time | O(n) space
7     public static int numberOfBinaryTreeTopologies(int n) {
8         List<Integer> cache = new ArrayList<Integer>();
9         cache.add(1);
10        for (int m = 1; m < n + 1; m++) {
11            int numberOfTrees = 0;
12            for (int leftTreeSize = 0; leftTreeSize < m; leftTreeSize++) {
13                int rightTreeSize = m - 1 - leftTreeSize;
14                int numberOfLeftTrees = cache.get(leftTreeSize);
15                int numberOfRightTrees = cache.get(rightTreeSize);
16                numberOfTrees += numberOfLeftTrees * numberOfRightTrees;
17            }
18            cache.add(numberOfTrees);
19        }
20        return cache.get(n);
21    }
22 }
23
```

Solution 1

Solution 2

Solution 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 }
```

```
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.