

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 // O(n^2) time | O(n) space
6 func NumberOfBinaryTreeTopologies(n int) int {
7     cache := []int{1}
8     for m := 1; m < n+1; m++ {
9         numberOfTrees := 0
10        for leftTreeSize := 0; leftTreeSize < m; leftTreeSize++ {
11            rightTreeSize := m - 1 - leftTreeSize
12            numberOfLeftTrees := cache[leftTreeSize]
13            numberOfRightTrees := cache[rightTreeSize]
14            numberOfTrees += numberOfLeftTrees * numberOfRightTrees
15        }
16        cache = append(cache, numberOfTrees)
17    }
18    return cache[n]
19 }
20
```

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

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

```
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.