

## Problem Challenge 3

We'll cover the following

- Count of Structurally Unique Binary Search Trees (hard)
- Try it yourself

### Count of Structurally Unique Binary Search Trees (hard) #

Given a number 'n', write a function to return the count of structurally unique Binary Search Trees (BST) that can store values 1 to 'n'.

**Example 1:**


```
Input: 2
Output: 2
Explanation: As we saw in the previous problem, there are 2 unique BSTs storing numbers from 1-2.
```


**Example 2:**


```
Input: 3
Output: 5
Explanation: There will be 5 unique BSTs that can store numbers from 1 to 3.
```


### Try it yourself #

Try solving this question here:

 Java

 Python3

 JS


 C++

```
1 class TreeNode:
2     def __init__(self, val):
3         self.val = val
4         self.left = None
5         self.right = None
6
7
8     def count_trees(n):
9         count = -1
10        # TODO: Write your code here
11        return count
12
13
14    def main():
15        print("Total trees: " + str(count_trees(2)))
16        print("Total trees: " + str(count_trees(3)))
17
18
19    main()
20
```

Run

Save

Reset



[← Back](#)

[Next →](#)

