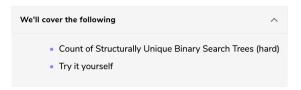


Problem Challenge 3



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:

```
Python3
                           JS JS
                                       G C++
  👙 Java
       import java.util.*;
      class TreeNode {
        int val;
TreeNode left;
        TreeNode right;
        TreeNode(int x) {
          val = x;
  13 class CountUniqueTrees {
        public int countTrees(int n) {
        public static void main(String[] args) {
          CountUniqueTrees ct = new CountUniqueTrees();
           int count = ct.countTrees(3);
           System.out.print("Total trees: " + count);
   Run
                                                                                            Save Reset []
 ← Back
                                                                                                         Next →
                                                                                     Solution Review: Problem Challenge 3
Solution Review: Problem Challenge 2
                                                                                                ✓ Mark as Completed
```