Problem Challenge 2

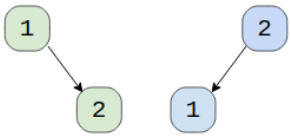
We'll cover the following

Structurally Unique Binary Search Trees (hard)

Given a number 'n', write a function to return all structurally unique Binary Search Trees (BST) that can store values 1 to 'n'?

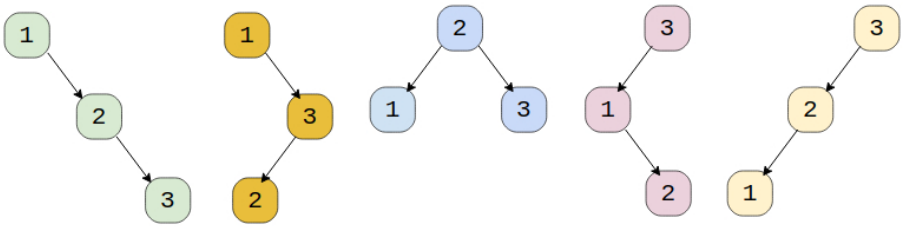
Example 1:

```
Input: 2
Output: List containing root nodes of all structurally unique BSTs.
Explanation: Here are the 2 structurally unique BSTs storing all numbers from 1 to 2:
```



Example 2:

```
Input: 3
Output: List containing root nodes of all structurally unique BSTs.
Explanation: Here are the 5 structurally unique BSTs storing all numbers from 1 to 3:
```



Try it yourself

Try solving this question here:

Java | Python3 | JS | C++

```python
class TreeNode:
  def __init__(self, val):
    self.val = val
    self.left = None
    self.right = None

def find_unique_trees(n):
  result = []
  # TODO: Write your code here
  return result
```

```
12
13
14 def main():
15   print("Total trees: " + str(len(find_unique_trees(2))))
16   print("Total trees: " + str(len(find_unique_trees(3))))
17
18
19 main()
20
```

Run Save Reset

Back

Solution Review: Problem Challenge 1

Next

Solution Review: Problem Challenge 2

Completed