Problem Challenge 2

We'll cover the following

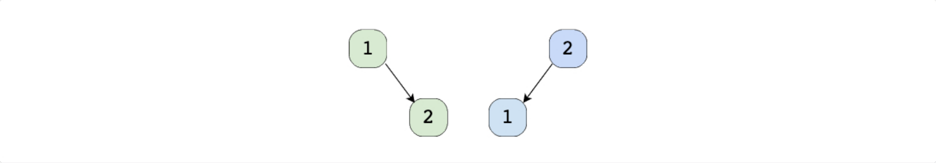
- Structurally Unique Binary Search Trees (hard)
- Try it yourself

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'?

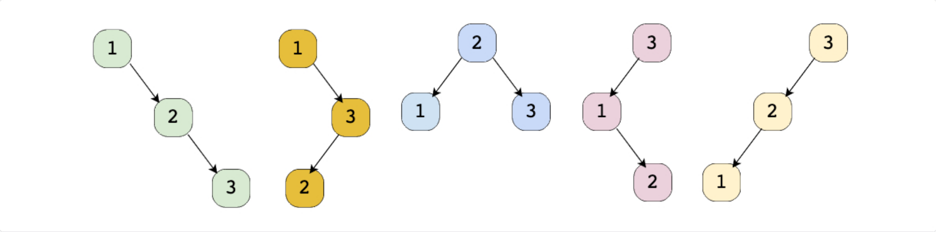
Example 1:

```
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++

```
import java.util.*;

class TreeNode {
  int val;
  TreeNode left;
  TreeNode right;

  TreeNode(int x) {
    val = x;
  }
};

class UniqueTrees {
  public static List<TreeNode> findUniqueTrees(int n) {
    List<TreeNode> result = new ArrayList<>();
    // TODO: Write your code here
    return result;
  }

  public static void main(String[] args) {
    List<TreeNode> result = UniqueTrees.findUniqueTrees(2);
    System.out.print("Total trees: " + result.size());
  }
}

```

Run Save Reset

← Back
Solution Review: Problem Challenge 1

Next →
Solution Review: Problem Challenge 2

☑ Mark as Completed

Report an Issue Ask a Question