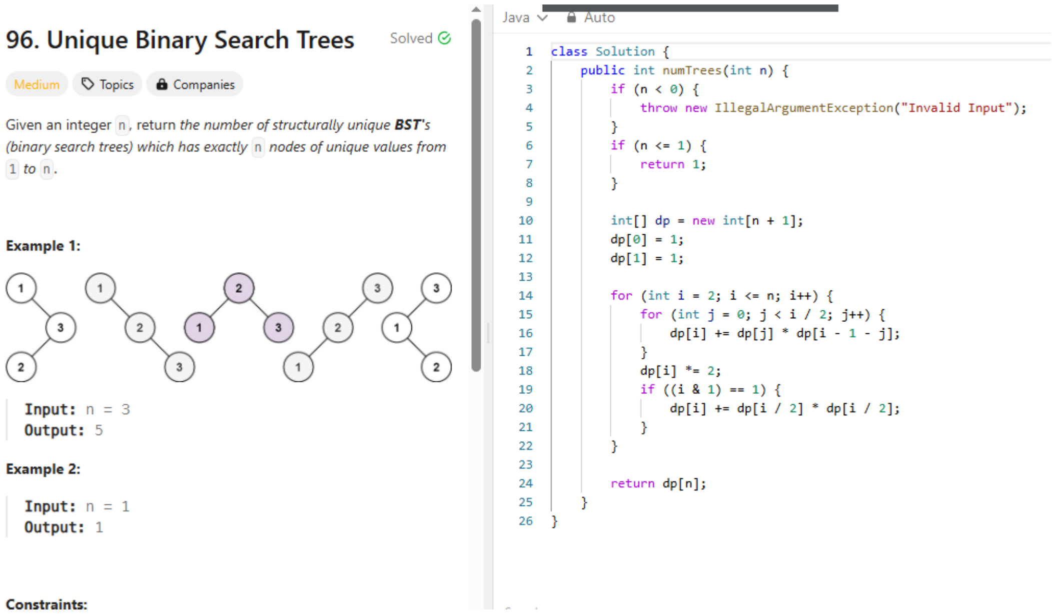
ADVANCED CODING 2 ASSIGNMENT 4

K.Srikar

Vu21csen0300058

Unique Binary search Trees:

Code:



96. Unique Binary Search Trees

Solved

Medium · Topics · Companies

Given an integer `n`, return *the number of structurally unique* ***BST'****s (binary search trees) which has exactly* `n` *nodes of unique values from* `1` *to* `n`.

Example 1:

```
Input: n = 3
Output: 5
```

Example 2:

```
Input: n = 1
Output: 1
```

Constraints:

Java · Auto

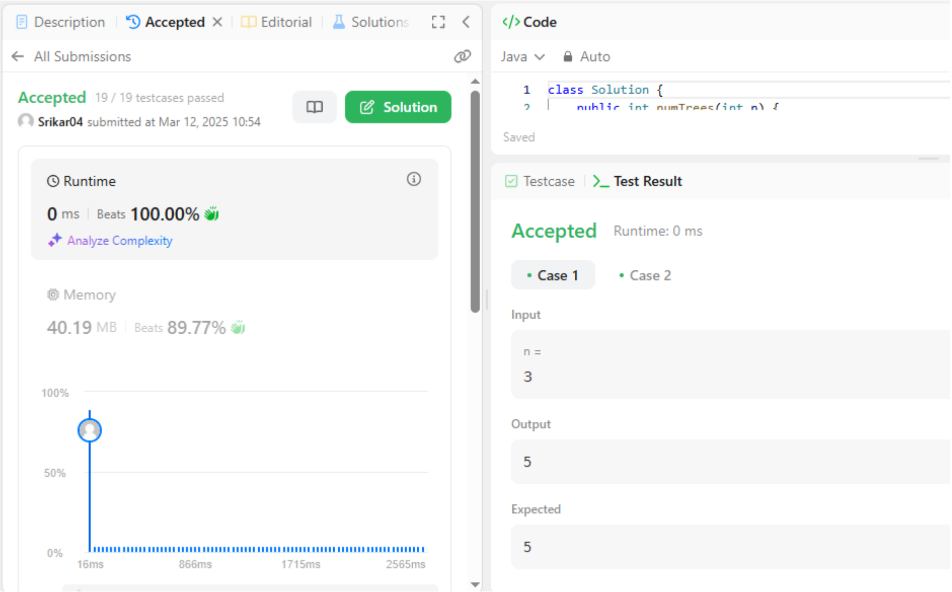
```java
class Solution {
    public int numTrees(int n) {
        if (n < 0) {
            throw new IllegalArgumentException("Invalid Input");
        }
        if (n <= 1) {
            return 1;
        }

        int[] dp = new int[n + 1];
        dp[0] = 1;
        dp[1] = 1;

        for (int i = 2; i <= n; i++) {
            for (int j = 0; j < i / 2; j++) {
                dp[i] += dp[j] * dp[i - 1 - j];
            }
            dp[i] *= 2;
            if ((i & 1) == 1) {
                dp[i] += dp[i / 2] * dp[i / 2];
            }
        }

        return dp[n];
    }
}
```

Output:



Accepted 19 / 19 testcases passed

Srikar04 submitted at Mar 12, 2025 10:54

Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

40.19 MB | Beats 89.77%

Code

Java · Auto

```java
class Solution {
    public int numTrees(int n) {
```

Saved

Accepted Runtime: 0 ms

Case 1 · Case 2

Input

n =

3

Output

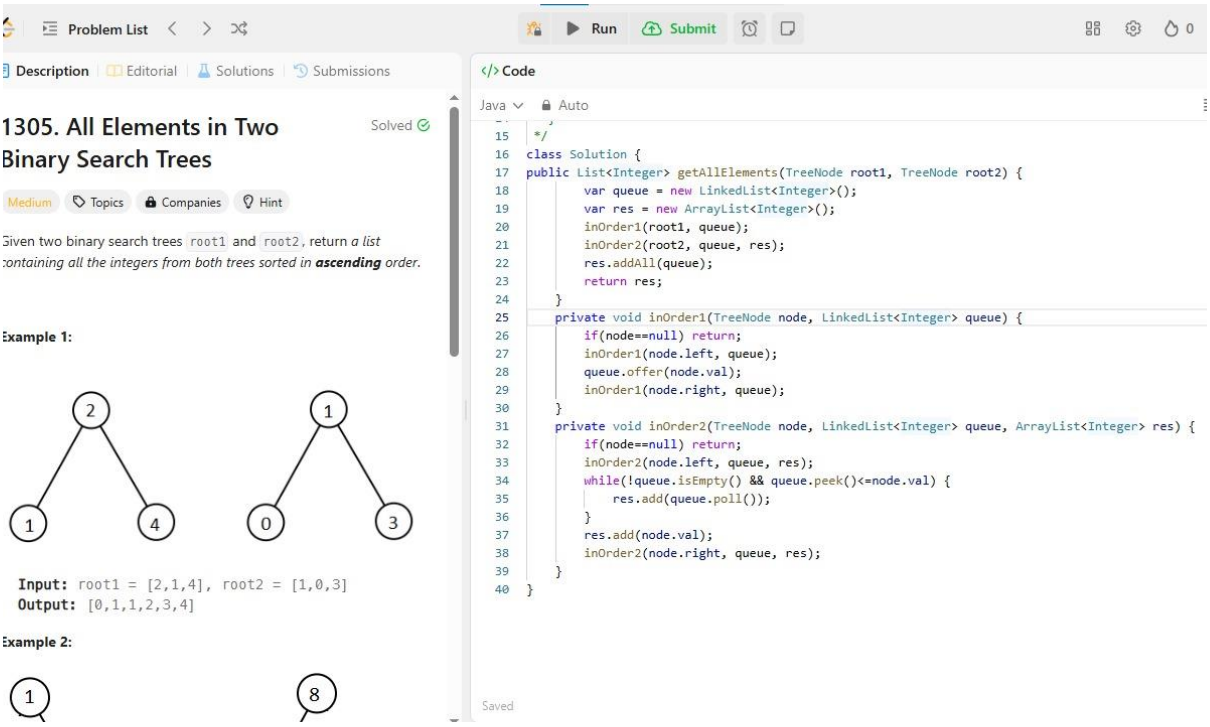
5

Expected

5

All elements in two binary search Trees:

Code:



1305. All Elements in Two Binary Search Trees

Given two binary search trees `root1` and `root2`, return *a list containing all the integers from both trees sorted in* ***ascending*** *order.*

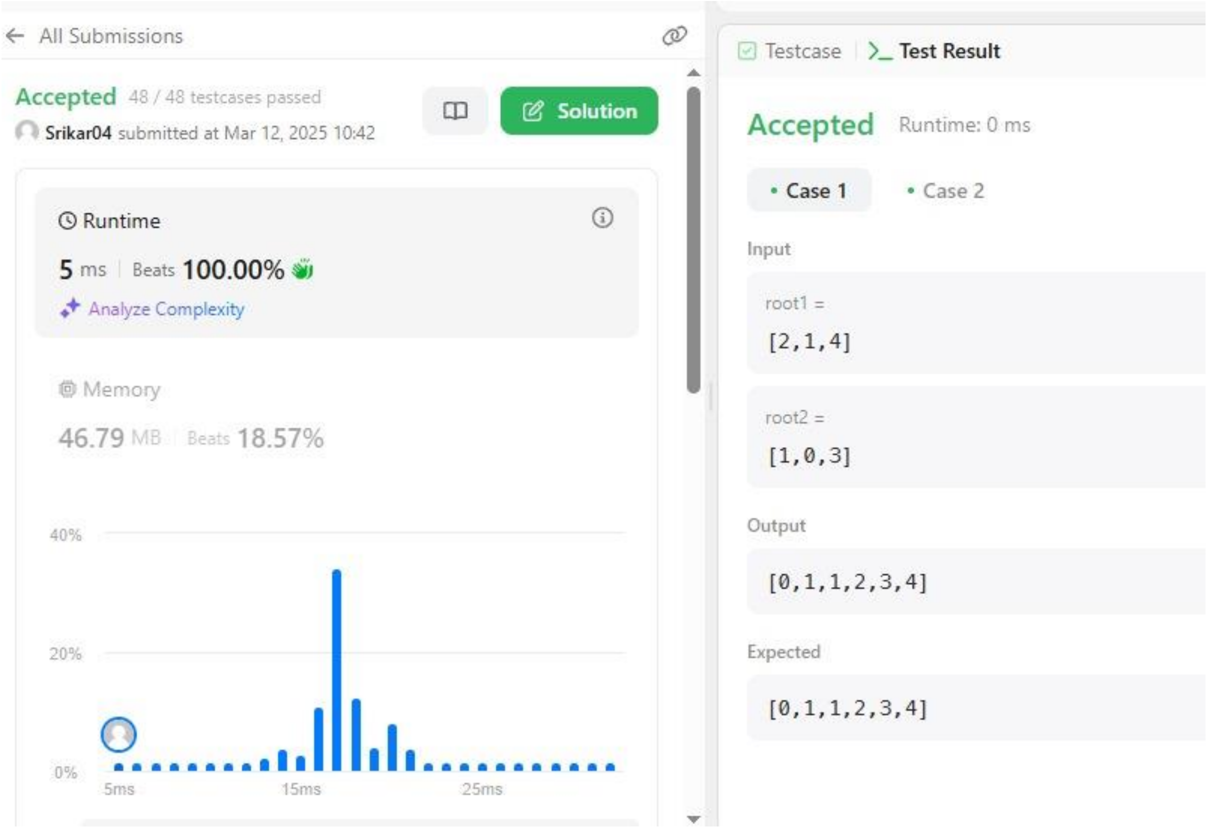
Example 1:

```
Input: root1 = [2,1,4], root2 = [1,0,3]
Output: [0,1,1,2,3,4]
```

Example 2:

```
 */
class Solution {
public List<Integer> getAllElements(TreeNode root1, TreeNode root2) {
        var queue = new LinkedList<Integer>();
        var res = new ArrayList<Integer>();
        inOrder1(root1, queue);
        inOrder2(root2, queue, res);
        res.addAll(queue);
        return res;
    }
    private void inOrder1(TreeNode node, LinkedList<Integer> queue) {
        if(node==null) return;
        inOrder1(node.left, queue);
        queue.offer(node.val);
        inOrder1(node.right, queue);
    }
    private void inOrder2(TreeNode node, LinkedList<Integer> queue, ArrayList<Integer> res) {
        if(node==null) return;
        inOrder2(node.left, queue, res);
        while(!queue.isEmpty() && queue.peek()<=node.val) {
            res.add(queue.poll());
        }
        res.add(node.val);
        inOrder2(node.right, queue, res);
    }
}
```

Output:



Accepted 48 / 48 testcases passed

Srikar04 submitted at Mar 12, 2025 10:42

Runtime

5 ms | Beats 100.00%

Memory

46.79 MB | Beats 18.57%

Accepted Runtime: 0 ms

• Case 1 • Case 2

Input

root1 =

```
[2,1,4]
```

root2 =

```
[1,0,3]
```

Output

```
[0,1,1,2,3,4]
```

Expected

```
[0,1,1,2,3,4]
```