

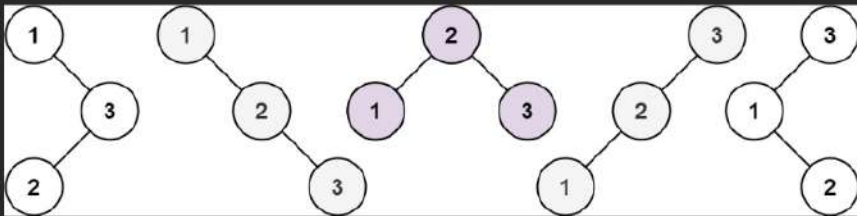
95. Unique Binary Search Trees II

Medium 6.9K 444

Companies

Given an integer n , return all the structurally unique **BST's** (binary search trees), which has exactly n nodes of unique values from 1 to n . Return the answer in **any order**.

Example 1:



Input: $n = 3$

Output: $[[1, \text{null}, 2, \text{null}, 3], [1, \text{null}, 3, 2], [2, 1, 3], [3, 1, \text{null}, \text{null}, 2], [3, 2, \text{null}, 1]]$

Example 2:

Input: $n = 1$

Output: $[[1]]$

Constraints:

i Java • Auto

```

15 //
16 class Solution {
17     public List<TreeNode> generateTrees(int n) {
18         List<TreeNode>[] dp = new ArrayList[n+1][n+1];
19         return memo(1, n, dp);
20     }
21     List<TreeNode> memo(int s, int e, List<TreeNode>[] dp) {
22         if (s > e) {
23             List<TreeNode> a = new ArrayList<>();
24             a.add(null);
25             return a;
26         }
27         if (dp[s][e] != null) return dp[s][e];
28         dp[s][e] = new ArrayList<>();
29         for (int i = s; i <= e; i++) {
30             List<TreeNode> left = memo(s, i-1, dp);
31             List<TreeNode> right = memo(i+1, e, dp);
32             for (TreeNode l : left)
33                 for (TreeNode r : right)

```

Testcase Result

Accepted Runtime: 0 ms

• Case 1 • Case 2

Input

$n =$

3

Output

$[[1, \text{null}, 2, \text{null}, 3], [1, \text{null}, 3, 2], [2, 1, 3], [3, 1, \text{null}, \text{null}, 2], [3, 2, \text{null}, 1]]$

Console

Run

Submit

Description

🔒 Editorial

Solutions (1.6K)

Submissions

89. Gray Code

Medium



👍 2K

💬 2.6K



🔒 Companies

An ***n*-bit gray code sequence** is a sequence of 2^n integers where:

- Every integer is in the **inclusive** range $[0, 2^n - 1]$,
- The first integer is 0 ,
- An integer appears **no more than once** in the sequence,
- The binary representation of every pair of **adjacent** integers differs by **exactly one bit**, and
- The binary representation of the **first** and **last** integers differs by **exactly one bit**.

Given an integer n , return *any valid ***n*-bit gray code sequence***.

Example 1:

Input: $n = 2$

Output: $[0, 1, 3, 2]$

Explanation:

The binary representation of $[0, 1, 3, 2]$ is $[00, 01, 11, 10]$.

– 00 and 01 differ by one bit

– 01 and 11 differ by one bit

– 11 and 10 differ by one bit

– 10 and 00 differ by one bit

$[0, 2, 3, 1]$ is also a valid gray code sequence, whose binary representation is $[00, 10, 11, 01]$.

– 00 and 10 differ by one bit

i Java | • Auto

```
1 class Solution {
2     public List<Integer> grayCode(int n) {
3         List<Integer> ans = new ArrayList<>();
4         int num = 0;
5         for (int i = 0; i < (1 << n); i++) {
6             // Compute the XOR of the current index i with (i & -i) to generate the next
        gray code.
7             num ^= i & (-i);
8             // Add the generated gray code to the answer list.
9             ans.add(num);
10        }
11        return ans;
12    }
13 }
```

Testcase

Result

Accepted Runtime: 0 ms

• Case 1

• Case 2

Input

 $n =$

2

Output

 $[0, 1, 3, 2]$

Expected

Console



Run

Submit