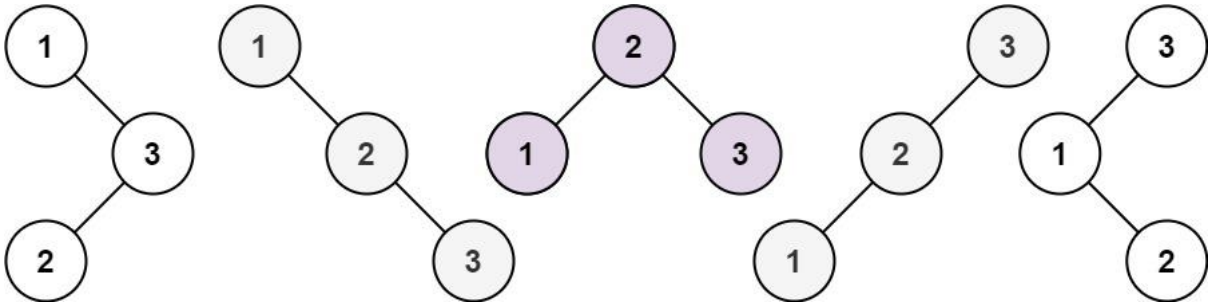
95. Unique Binary Search Trees II

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,null,2,null,3],[1,null,3,2],[2,1,3],[3,1,null,null,2],[3,2,null,1]]

Example 2:

Input: n = 1

Output: [[1]]

Constraints:

- 1 <= n <= 8

```
# Definition for a binary tree node.
# class TreeNode(object):
#     def __init__(self, val=0, left=None, right=None):
#         self.val = val
#         self.left = left
#         self.right = right
class Solution(object):
    def generateTrees(self, n):
        def rec(start, end):
            if(start > end):
                return [None]
            if(start == end):
                return [TreeNode(start)]

            ret_list = []

            for i in range(start, end+1):
                left = rec(start,i-1)
                right = rec(i+1,end)
                for pair in product(left,right):

ret_list.append(TreeNode(i,pair[0],pair[1]))
            return ret_list
        res = rec(1,n)
        return res
```