# Definition for a binary tree node.

# class TreeNode:

# def \_\_init\_\_(self, val=0, left=None, right=None):

# self.val = val

# self.left = left

# self.right = right

class Solution:

def generateTrees(self, n: int) -> List[Optional[TreeNode]]:

if n == 0:

return []

memo = {}

def generate\_trees(start, end):

if (start, end) in memo:

return memo[(start, end)]

trees = []

if start > end:

trees.append(None)

return trees

for root\_val in range(start, end + 1):

left\_trees = generate\_trees(start, root\_val - 1)

right\_trees = generate\_trees(root\_val + 1, end)

for left\_tree in left\_trees:

for right\_tree in right\_trees:

root = TreeNode(root\_val, left\_tree, right\_tree)

trees.append(root)

memo[(start, end)] = trees

return trees

return generate\_trees(1, n)