**Unique Binary Search Tree**

Given n, how many structurally unique **BST's** (binary search trees) that store values 1 ... n

boolean isPalindrome**(**LinkedListNode head**)** **{**

class Solution **{**

public int numTrees**(**int n**)** **{**

int **[]** cat **=** **new** int**[**n**+**1**];**

cat**[**0**]=**cat**[**1**]=**1**;**

**for(**int i**=**2**;**i**<=**n**;**i**++){**

**for(**int j **=** 1**;** j**<=**i**;** j**++){**

cat**[**i**]+=**cat**[**j**-**1**]\***cat**[**i**-**j**];**

**}**

**}**

**return** cat**[**n**];**

**}**

**}**

**}**

G(n) = F(1, n) + F(2, n) + ... + F(n, n).

G(0)=1, G(1)=1.

F(**i**, n) = G(**i-1**) \* G(**n-i**) 1 <= i <= n

G(n) = G(0) \* G(n-1) + G(1) \* G(n-2) + … + G(n-1) \* G(0)