1. Unique Binary Search Trees

Medium

Given *n*, how many structurally unique **BST’s** (binary search trees) that store values 1 … *n*?

**Example:**

Input: 3  
Output: 5  
Explanation:  
Given n = 3, there are a total of 5 unique BST's:  
  
 1 3 3 2 1  
 \ / / / \ \  
 3 2 1 1 3 2  
 / / \ \  
 2 1 2 3

**Solution**

Approach1

动态规划

设：

: n个节点，以节点 i 作为根节点的二叉树个数

: n个节点形成的BST个数

显然有：

$$
G(n) = \sum\_{i=1}^n F(i, n)\\
F(i, n) = G(i-1)\*G(n-i)
$$

合并得到：

初始化，递归计算

class Solution {  
public:  
 int numTrees(int n) {  
 int dp[n+1] = {0};  
 dp[0] = 1, dp[1] = 1;  
 for(int i = 2; i <= n; ++i){  
 for(int j = 1; j <= i; ++j)dp[i] += dp[j-1]\*dp[i-j];  
 }  
 return dp[n];  
 }  
};

Approach2

公式3的结果是卡特数