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

For example,  
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

思路：

动态规划的思想。使用动态规划的两个特征：

最优子结构，包含子问题的最优解

重叠子问题，反复求解相同的子问题

num[i] 中保存表示i个节点时的表示方式个数

节点个数为0和1时只有一种表示方式：num[0]=1 num[1]=1

节点个数为2时：

①1为根节点：左子树上有0个节点，右子树上有两个节点 num[0]\*num[1] ，相乘表示左右两个子树组成的总的子树的个数

② 2为根节点，左子树上有1个节点，右子树上有0个节点

num[1]\*num[0]

num[2] = ①+②

节点个数为3时：

①1为根节点，左子树0个，右子树2个 num[0]\*num[2]

②2为根节点，左子树1个，右子树1个，num[1]\*num[1]

③3为根节点，左子树2两个，右子树1个，num[2]\*num[1]

num[3]= ①+②+③

……