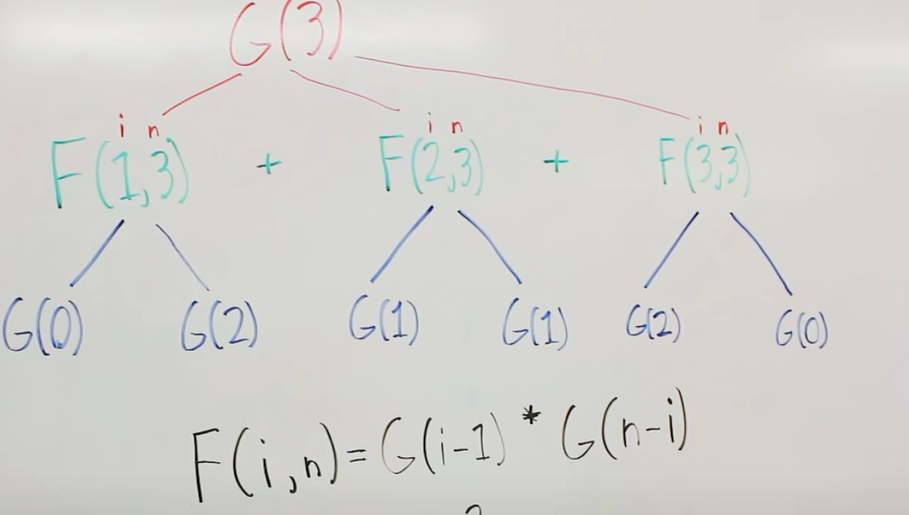


知识点：二叉搜索树左边的数必然小于root,右边的数必然大于root

这题是动态规划，catalan number



假设有n个数，我选其中一个数i为root，那么左边i-1个数就是子问题，他一共有a种可能，右边n-i-1个数也是子问题，一共有b中可能，两者相乘得到总共的可能性，则整个树的形态有 a \* b种。

同时用123构建的可能性与用234构建的可能性是一样的，

因此我们只关注左边有几个数

class Solution {

    public int numTrees(int n) {

        int[] catalan=new int[n+1];

        catalan[0]=1; 0个数 ，1种可能

        catalan[1]=1; 0个数， 1种可能

        for(int i=2;i<=n;i++){ //边界是等于n,因为catalan[0] 并不对应只有一个数1的情况，而是对应一个数都没有的情况

            catalan[i]=0;

            for(int j=0;j<i;j++){

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

            } 假设有4个数， 那么catalan[4]也就是我们的最终目的，4的分法有，1为节点，左边0个，右边3个，2,为节点，左边1个右边2个，3为节点左边2个右边1个，4为节点左边3个右边0个，**我们要让两者相乘**

        }

        return catalan[n];

    }

}