1. 题目
2. 题干

给定一个整数 n，求以 1 ... n 为节点组成的二叉搜索树有多少种？

1. 示例

输入: 3

输出: 5

解释:

给定 n = 3, 一共有 5 种不同结构的二叉搜索树:

1 3 3 2 1

\ / / / \ \

3 2 1 1 3 2

/ / \ \

2 1 2 3

1. 题解
2. 思路

二叉搜索树，树的根节点大于左子树中的所有值，小于右子树的所有值。关于树的题目，一般都要往递归和动态规划方面去思考。很明显，这个题目可以使用动态规划解决，n个数的二叉搜索树有多少种，可以从n=1开始迭代，迭代n次。每次迭代中，选取一个数作为根节点，分出左右子树。每个子树能够构成的二叉搜索树只和每个子树的节点个数相关，与节点的内容是没有关系的，因为每个子树都是有序的，二叉搜索树都是按照顺序排列的。因此，n次迭代中每个根节点能组成的二叉搜索树的种类为左子树的种类乘以右子树的种类。

1. 代码实现

Java：

class Solution {

    public int numTrees(int n) {

        int[] dp = new int[n+1];

        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];

    }

}