

96.不同的二叉搜索树

题目链接: <https://leetcode-cn.com/problems/unique-binary-search-trees/>

给定一个整数 n , 求以 $1 \dots n$ 为节点组成的二叉搜索树有多少种?

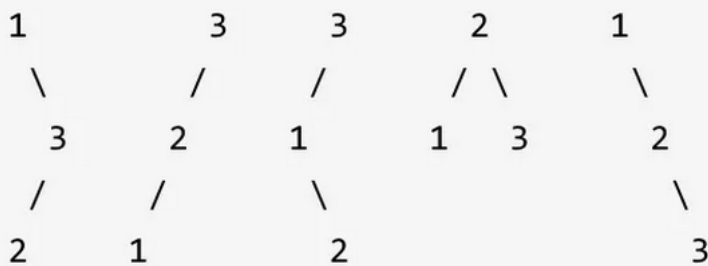
示例:

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:



$dp[3]$ 就是元素为1的头节点的搜索树的数量+元素为2的头节点的搜索树的数量+元素为3的头节点的搜索树的数量

元素为1头结点的搜索树的数量=右子树两个元素的搜索树的数量= $dp[2]$

元素为2头结点的搜索树的数量=左子树1个元素的搜索树的数量*右子树1个元素的搜索树的数量
= $dp[1]*dp[1]$

元素为3头结点的搜索树的数量=左子树2个元素的搜索树的数量= $dp[2]$

$dp[3]=dp[2]+dp[1]*dp[1]+dp[2]$

1、dp数组的下标及其定义:

$dp[i]$:表示以 $1 \dots i$ 节点组成搜索树的数量。

2、递推公式

$dp[i] += dp[i-j]*dp[j-1] \quad 1 \leq j \leq i$

3、dp数组初始化

$dp[0]=1, dp[1]=1, dp[2]=2$

4、从左向右遍历

```
1 class Solution {
2     public:
3         int numTrees(int n) {
```

```
4  if(n==1) return 1;
5      if(n==2) return 2;
6      vector<int> dp(n+1,0);
7      dp[0]=dp[1]=1;dp[2]=2;
8      for(int i=3;i<=n;i++){
9          for(int j=1;j<=i;j++){
10             dp[i]+=dp[j-1]*dp[i-j];
11         }
12     }
13     return dp[n];
14 }
15 };
```