0096. 不同的二叉搜索树

ITCharge 大约 2 分钟

- 标签：树、二叉搜索树、数学、动态规划、二叉树
- 难度：中等

题目链接

- 0096. 不同的二叉搜索树 - 力扣

题目大意

描述：给定一个整数 n。

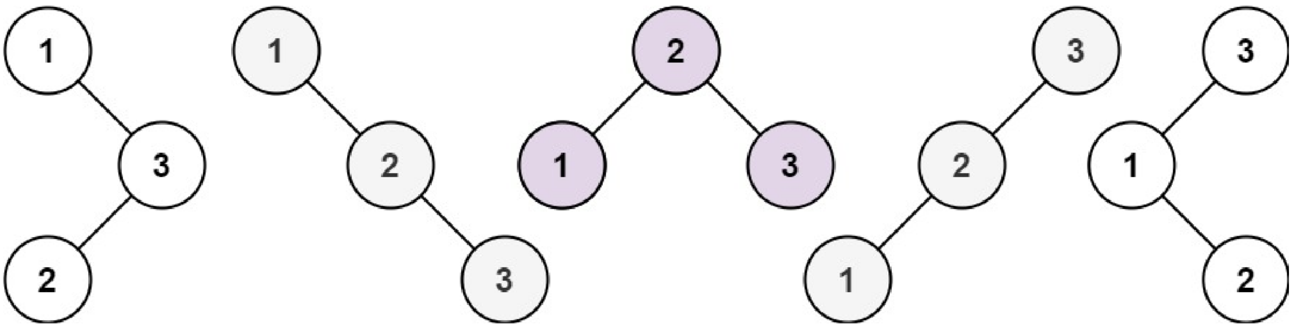
要求：求以 1 到 n 为节点构成的「二叉搜索树」有多少种？

说明：

- $1 \le n \le 19$。

示例：

- 示例 1：



```py
输入：n = 3
输出：5
```

- 示例 2：

```py
输入：n = 1
输出：1
```

解题思路

思路 1：动态规划

一棵搜索二叉树的左、右子树，要么也是搜索二叉树，要么就是空树。

如果定义 $f[i]$ 表示以 i 为根的二叉搜索树个数，定义 $g(i)$ 表示 i 个节点可以构成的二叉搜索树个数，则有：

- $g(i) = f(1) + f(2) + f(3) + \ldots + f(i)$。

其中当 i 为根节点时，则用 $(1, 2, \ldots, i - 1)$ 共 $i - 1$ 个节点去递归构建左子搜索二叉树，用 $(i + 1, i + 2, \ldots, n)$ 共 $n - i$ 个节点去递归构建右子搜索树。则有：

- $f(i) = g(i - 1) \times g(n - i)$。

综合上面两个式子 $\begin{cases} g(i) = f(1) + f(2) + f(3) + \ldots + f(i) \\ f(i) = g(i - 1) \times g(n - i) \end{cases}$ 可得出：

- $g(n) = g(0) \times g(n - 1) + g(1) \times g(n - 2) + \ldots + g(n - 1) \times g(0)$。

将 n 换为 i，可变为：

- $g(i) = g(0) \times g(i - 1) + g(1) \times g(i - 2) + \ldots + g(i - 1) \times g(0)$。

再转换一下，可变为：

- $g(i) = \sum_{1 \le j \le i} \{g(j - 1) \times g(i - j)\}$。

则我们可以通过动态规划的方法，递推求解 $g(i)$，并求解出 $g(n)$。具体步骤如下：

1. 划分阶段

按照根节点的编号进行阶段划分。

2. 定义状态

定义状态 $dp[i]$ 表示为： i 个节点可以构成的二叉搜索树个数。

3. 状态转移方程

$dp[i] = \sum_{1 \le j \le i} \{dp[j - 1] \times dp[i - j]\}$

4. 初始条件

- 0 个节点可以构成的二叉搜索树个数为 1（空树），即 $dp[0] = 1$。

5. 最终结果

根据我们之前定义的状态，$dp[i]$ 表示为： i 个节点可以构成的二叉搜索树个数。。 所以最终结果为 $dp[n]$。

思路 1：代码

```py
class Solution:
    def numTrees(self, n: int) -> int:
        dp = [0 for _ in range(n + 1)]
        dp[0] = 1
        for i in range(1, n + 1):
            for j in range(1, i + 1):
                dp[i] += dp[j - 1] * dp[i - j]
        return dp[n]
```

思路 1：复杂度分析

- **时间复杂度**：$O(n^2)$。
- **空间复杂度**：$O(n)$。

参考资料

- 【题解】[画解算法：96. 不同的二叉搜索树 - 不同的二叉搜索树]