入选理由

1. 二叉搜索树是最最适合练习分治的数据结构

标签

- 分治

难度

- 中等

题目地址（96. 不同的二叉搜索树）

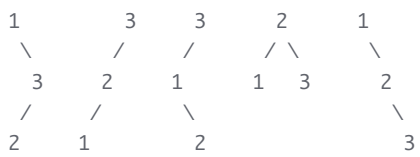
https://leetcode-cn.com/problems/unique-binary-search-trees/

题目描述

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?

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：



前置知识

- 二叉搜索树
- 分治

公司

- 阿里
- 腾讯
- 百度
- 字节

岗位信息

- 腾讯（广州）- 安卓 - 社招 - 三面

思路

这是一个经典的使用分治思路的题目。

对于数字 n ，我们可以 1- n 这样的离散整数分成左右两部分。我们不妨设其分别为 A 和 B。那么问题转化为 A 和 B 所能组成的 BST 的数量的笛卡尔积。而对于 A 和 B 以及原问题除了规模，没有不同，这不就是分治思路么？至于此，我们只需要考虑边界即可，边界很简单就是 n 小于等于 1 的时候，我们返回 1。

具体来说：

- 生成一个[1:n + 1] 的数组
- 我们遍历一次数组，对于每一个数组项，我们执行以下逻辑
- 对于每一项，我们都假设其是断点。断点左侧的是 A，断点右侧的是 B。
- 那么 A 就是 i - 1 个数， 那么 B 就是 n - i 个数
- 我们递归，并将 A 和 B 的结果相乘即可。

> 其实我们发现，题目的答案只和 n 有关，和具体 n 个数的具体组成，只要是有序数组即可。

题目没有明确 n 的取值范围，我们试一下暴力递归。

代码（Python3）：

```
class Solution:
    def numTrees(self, n: int) -> int:
        if n <= 1:
            return 1
        res = 0
```

```
        for i in range(1, n + 1):
            res += self.numTrees(i - 1) * self.numTrees(n - i)
        return res
```

上面的代码会超时，并没有栈溢出，因此我们考虑使用 hashmap 来优化，代码见下方代码区。

关键点解析

- 分治法
- 笛卡尔积
- 记忆化递归

代码

语言支持：Python3, CPP

Python3 Code:

```
class Solution:
    visited = dict()

    def numTrees(self, n: int) -> int:
        if n in self.visited:
            return self.visited.get(n)
        if n <= 1:
            return 1
        res = 0
        for i in range(1, n + 1):
            res += self.numTrees(i - 1) * self.numTrees(n - i)
        self.visited[n] = res
        return res
```

CPP Code:

```
class Solution {
    vector<int> visited;
    int dp(int n) {
        if (visited[n]) return visited[n];
        int ans = 0;
        for (int i = 0; i < n; ++i) ans += dp(i) * dp(n - i - 1);
        return visited[n] = ans;
    }
```

```
public:
    int numTrees(int n) {
        visited.assign(n + 1, 0);
        visited[0] = 1;
        return dp(n);
    }
};
```

复杂度分析

- 时间复杂度：一层循环是 N，另外递归深度是 N，因此总的时间复杂度是 $O(N^2)$
- 空间复杂度：递归的栈深度和 visited 的大小都是 N，因此总的空间复杂度为 $O(N)$

相关题目

- 95.unique-binary-search-trees-ii

更多题解可以访问我的 LeetCode 题解仓库：https://github.com/azl397985856/leetcode 。 目前已经 30K star 啦。

关注公众号力扣加加，努力用清晰直白的语言还原解题思路，并且有大量图解，手把手教你识别套路，高效刷题。

上一页 下一页