不同路径 I

题述

62. 不同路径

难度 中等 1361

一个机器人位于一个 `m x n` 网格的左上角 （起始点在下图中标记为 “Start” ）。

机器人每次只能向下或者向右移动一步。机器人试图达到网格的右下角 （在下图中标记为 “Finish” ）。

问总共有多少条不同的路径?

示例 1:


```
输入：m = 3, n = 7
输出：28
```

示例 2:

```
输入：m = 3, n = 2
输出：3
解释：
从左上角开始，总共有 3 条路径可以到达右下角。
1. 向右 -> 向下 -> 向下
2. 向下 -> 向下 -> 向右
3. 向下 -> 向右 -> 向下
```

思路

深度优先搜索

回想起学习过的图论知识，可以使用深度优先搜索来遍历整个图，来枚举路径总数

动态规划

1、确定dp数组（dp table）以及下标的含义

dp[i] [j]： 表示从起点（0 ， 0）出发，到(i, j) 有dp[i] [j]条不同的路径。

2、确定递推公式

要求dp[i] [j]，只能有两个方向来推导出来，即dp[i - 1] [j] 和 dp[i] [j - 1]。

```
dp[i][j] = dp[i - 1][j] + dp[i][j - 1]
```

3、dp数组的初始化

首先dp[i] [0]一定都是1，因为从(0, 0)的位置到(i, 0)的路径只有一条，那么dp[0] [j]也同理。

```
dp[i][0] = 1
dp[0][j] = 1
```

4、确定遍历顺序

dp[i] [j] = dp[i - 1] [j] + dp[i] [j - 1]，dp[i] [j]都是从其上方和左方推导而来，那么从左到右一层一层遍历就可以了。

5、举例推导dp数组

题解

Python

深度优先搜索

```python
class Solution:
    def dfs(self,i,j,m,n):
        # 图的深度优先搜索
        if i > m or j >n:
            return 0
        if i == m and j == n:
            return 1    # 找到了一条路径
        return self.dfs(i+1,j,m,n) + self.dfs(i,j+1,m,n)    # 递归
    def uniquePaths(self, m: int, n: int) -> int:
        return self.dfs(1,1,m,n)    #递归
```

动态规划

```python
class Solution:
    def uniquePaths(self, m: int, n: int) -> int:
        # 动态规划
        # dp[i][j] = dp[i - 1][j] + dp[i][j - 1]

        # 初始化dp数组
        dp = [[1 for i in range(n)] for j in range(m)]
        for i in range(1,m):
            for j in range(1,n):
                dp[i][j] = dp[i - 1][j] + dp[i][j - 1]
        return dp[m-1][n-1]
```

提交结果	执行用时	内存消耗	语言	提交时间
通过	44 ms	15 MB	Python3	2022/04/09 10:17
超出时间限制	N/A	N/A	Python3	2022/04/09 08:56

思考