

# 0064. 最小路径和

👤 ITCharge 🕒 大约 2 分钟

- 标签：数组、动态规划、矩阵
- 难度：中等

## 题目链接

- [0064. 最小路径和 - 力扣](#)

## 题目大意

**描述：** 给定一个包含非负整数的  $m \times n$  大小的网格 *grid*。

**要求：** 找出一条从左上角到右下角的路径，使得路径上的数字总和为最小。

**说明：**

- 每次只能向下或者向右移动一步。
- $m == \text{grid.length}$ 。
- $n == \text{grid}[i].length$ 。
- $1 \leq m, n \leq 200$ 。
- $0 \leq \text{grid}[i][j] \leq 100$ 。

**示例：**

- 示例 1：

|   |   |   |
|---|---|---|
| 1 | 3 | 1 |
| 1 | 5 | 1 |
| 4 | 2 | 1 |

py

输入: `grid = [[1,3,1],[1,5,1],[4,2,1]]`

输出: 7

解释: 因为路径 `1→3→1→1→1` 的总和最小。

- 示例 2:

py

输入: `grid = [[1,2,3],[4,5,6]]`

输出: 12

## 解题思路

### 思路 1: 动态规划

#### 1. 划分阶段

按照路径的结尾位置（行位置、列位置组成的二维坐标）进行阶段划分。

#### 2. 定义状态

定义状态  $dp[i][j]$  为: 从左上角到达  $(i, j)$  位置的最小路径和。

#### 3. 状态转移方程

当前位置  $(i, j)$  只能从左侧位置  $(i, j - 1)$  或者上方位置  $(i - 1, j)$  到达。为了使得从左上角到达  $(i, j)$  位置的最小路径和最小, 应从  $(i, j - 1)$  位置和  $(i - 1, j)$  位置选择路径和最小的位置达到  $(i, j)$ 。

即状态转移方程为:  $dp[i][j] = \min(dp[i][j - 1], dp[i - 1][j]) + grid[i][j]$ 。

#### 4. 初始条件

- 当左侧和上方是矩阵边界时 (即  $i = 0, j = 0$ ) ,  $dp[i][j] = grid[i][j]$ 。
- 当只有左侧是矩阵边界时 (即  $i \neq 0, j = 0$ ) , 只能从上方到达,  $dp[i][j] = dp[i - 1][j] + grid[i][j]$ 。
- 当只有上方是矩阵边界时 (即  $i = 0, j \neq 0$ ) , 只能从左侧到达,  $dp[i][j] = dp[i][j - 1] + grid[i][j]$ 。

## 5. 最终结果

根据状态定义，最后输出  $dp[rows - 1][cols - 1]$ （即从左上角到达  $(rows - 1, cols - 1)$  位置的最小路径和）即可。其中  $rows$ 、 $cols$  分别为  $grid$  的行数、列数。

## 思路 1：代码

```
class Solution:
    def minPathSum(self, grid: List[List[int]]) -> int:
        rows, cols = len(grid), len(grid[0])
        dp = [[0 for _ in range(cols)] for _ in range(rows)]

        dp[0][0] = grid[0][0]

        for i in range(1, rows):
            dp[i][0] = dp[i - 1][0] + grid[i][0]

        for j in range(1, cols):
            dp[0][j] = dp[0][j - 1] + grid[0][j]

        for i in range(1, rows):
            for j in range(1, cols):
                dp[i][j] = min(dp[i][j - 1], dp[i - 1][j]) + grid[i][j]

        return dp[rows - 1][cols - 1]
```

## 思路 1：复杂度分析

- 时间复杂度： $O(m * n)$ ，其中  $m$ 、 $n$  分别为  $grid$  的行数和列数。
- 空间复杂度： $O(m * n)$ 。