

剑指offer47.礼物的最大价值

题目描述

在一个 $m \times n$ 的棋盘的每一格都放有一个礼物，每个礼物都有一定的价值（价值大于 0）。你可以从棋盘的左上角开始拿格子里的礼物，并每次向右或者向下移动一格、直到到达棋盘的右下角。给定一个棋盘及其上面的礼物的价值，请计算你最多能拿到多少价值的礼物？

示例 1:

输入:

```
[
  [1,3,1],
  [1,5,1],
  [4,2,1]
]
```

输出: 12

解释: 路径 $1 \rightarrow 3 \rightarrow 5 \rightarrow 2 \rightarrow 1$ 可以拿到最多价值的礼物

来源: 力扣 (LeetCode)

链接: <https://leetcode-cn.com/problems/li-wu-de-zui-da-jie-zhi-lcof>

解题思路

- 这是典型的动态规划题目，按步骤解决：
- 定义状态，
 - 用 $f(i,j)$ 表示到达坐标 (i,j) 的格子时能够拿到的礼物的最大价值
- 寻找状态转移方程，
 - 到达格子 (i,j) 有两种方法，从左边 $(i,j-1)$ 或者上边 $(i-1,j)$ 到达， $f(i,j)$ 应该等于左边格子和上边格子礼物价值的最大值再加上格子 (i,j) 礼物的价值
 - $f(i,j) = \max(f(i,j-1), f(i-1,j)) + \text{grid}(i,j)$
- 状态初始化
 - 左上角为起点，没有方法可以到达，则 $f(0,0) = \text{grid}(0,0)$
 - 最上边一行只能从左边到达，则有 $f(0,j) = f(0,j-1) + \text{grid}(0,j)$
 - 最左边一列只能从上边到达，则有 $f(i,0) = f(i-1,0) + \text{grid}(i,0)$
- 返回结果
 - 右下角元素，即 $f(\text{row}-1, \text{col}-1)$

代码实现

```
class Solution {
public:
    int maxValue(vector<vector<int>>& grid) {
        int row=grid.size();
        int col=grid[0].size();
        vector<vector<int>> dp(row,vector<int>(col,0));
        //初始化
        dp[0][0]=grid[0][0];
```

```
    for(size_t i=1;i<row;++i)
    {
        dp[i][0]=dp[i-1][0]+grid[i][0];
    }
    for(size_t j=1;j<col;++j)
    {
        dp[0][j]=dp[0][j-1]+grid[0][j];
    }
    //状态方程
    for(size_t i=1;i<row;++i)
    {
        for(size_t j=1;j<col;++j)
        {
            dp[i][j]=max(dp[i-1][j],dp[i][j-1])+grid[i][j];
        }
    }
    //返回结果
    return dp[row-1][col-1];
}

};
```