# 题目

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”）。

现在考虑网格中有障碍物。那么从左上角到右下角将会有多少条不同的路径？

网格中的障碍物和空位置分别用1和0来表示。

示例 1：

输入：obstacleGrid = [[0,0,0],[0,1,0],[0,0,0]]

输出：2

解释：3x3 网格的正中间有一个障碍物。

从左上角到右下角一共有 2 条不同的路径：

1. 向右 -> 向右 -> 向下 -> 向下

2. 向下 -> 向下 -> 向右 -> 向右

示例 2：

输入：obstacleGrid = [[0,1],[0,0]]

输出：1

提示：

m == obstacleGrid.length

n == obstacleGrid[i].length

1 <= m, n <= 100

obstacleGrid[i][j] 为 0 或 1

# 分析

class Solution {

public:

int uniquePathsWithObstacles(vector<vector<int>>& obstacleGrid) {

int m = obstacleGrid.size(), n = obstacleGrid[0].size();

vector<int> vec(n);

vec[0] = (obstacleGrid[0][0] == 0);

for (int i = 0; i < m; ++i) {

for (int j = 0; j < n; j++) {

if (obstacleGrid[i][j] == 1) {

vec[j] = 0;

continue;

}

if (j - 1 >= 0 && obstacleGrid[i][j - 1] == 0) {

vec[j] += vec[j - 1];

}

}

}

return vec.back();

}

};