

给定一个包含非负整数的 $m \times n$ 网格 `grid`，请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。

分析：

1. 对此类二维矩阵的dp, 状态转移方程在题目中已经给出，例如，本题中的每次只能向下或者向右移动一步。
2. 在计算dp矩阵时，只需根据给定的状态转移方程计算即可。
3. 对base case的计算，也是根据题目给定的状态转移方程，进行判断。

```
class Solution {
public:
    int minPathSum(std::vector<std::vector<int>>& grid) {
        int row = grid.size(), col = 0;
        if (row == 0) {
            return 0;
        } else {
            col = grid[0].size();
        }
        std::vector<std::vector<int>> dp =
            std::vector<std::vector<int>>(row + 1, std::vector<int>(col + 1,
0));
        dp[0][0] = grid[0][0];

        // base case
        for (int i = 1; i < row; i++) {
            dp[i][0] = dp[i - 1][0] + grid[i][0];
        }

        for (int i = 1; i < col; i++) {
            dp[0][i] = dp[0][i - 1] + grid[0][i];
        }

        // 计算dp
        for (int i = 1; i < row; i++) {
            for (int j = 1; j < col; j++) {
                dp[i][j] = std::min(dp[i - 1][j], dp[i][j - 1]) + grid[i][j];
            }
        }

        return dp[row - 1][col - 1];
    }
};
```