# 题目

Given a m x n grid filled with non-negative numbers, find a path from top left to bottom right which minimizes the sum of all numbers along its path.

Note: You can only move either down or right at any point in time.

# 解题思路

递推公式:

dp[i][j] = min(dp[i][j-1], dp[i - 1][j]) + grid[i][j]

另外，dp table 最左边的一列需要依次向后累加，既

for (vector<vector<int>>::size\_type i = 1; i < m; i++)

table[i][0] = grid[i][0] + table[i - 1][0];

同理 最上边 这一行也需要依次向后累加。

for (vector<int>::size\_type i = 1; i < n; i++)

table[0][i] = grid[0][i] + table[0][i - 1];

最后，dp table 的（0,0）坐标点的值需要复制 grid (0,0)坐标点的值。

# 代码

int minPathSum(const vector<vector<int>>& grid) {

vector<vector<int>>::size\_type m = grid.size();

vector<int>::size\_type n = grid[0].size();

vector<vector<int>> table(m, vector<int>(n));

table[0][0] = grid[0][0];

for (vector<vector<int>>::size\_type i = 1; i < m; i++)

table[i][0] = grid[i][0] + table[i - 1][0];

for (vector<int>::size\_type i = 1; i < n; i++)

table[0][i] = grid[0][i] + table[0][i - 1];

for (vector<vector<int>>::size\_type i = 1; i < m; i++) {

for (vector<int>::size\_type j = 1; j < n; j++) {

table[i][j] = min(table[i][j - 1], table[i - 1][j]) + grid[i][j];

}

}

return table[m - 1][n - 1];

}

## 优化了DP table 以后

int minPathSum(vector<vector<int>>& grid) {

for (unsigned int i = 1; i < grid.size(); i++)

grid[i][0] += grid[i - 1][0];

for (unsigned int i = 1; i < grid[0].size(); i++)

grid[0][i] += grid[0][i - 1];

for (unsigned int i = 1; i < grid.size(); i++) {

for (unsigned int j = 1; j < grid[0].size(); j++) {

grid[i][j] += min(grid[i][j - 1], grid[i - 1][j]);

}

}

return grid[grid.size() - 1][grid[0].size() - 1];

}