1. Minimum Path Sum

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.

**Example:**

Input:  
[  
 [1,3,1],  
 [1,5,1],  
 [4,2,1]  
]  
Output: 7  
Explanation: Because the path 1→3→1→1→1 minimizes the sum.

**解**

递归求解，每次只有两个方向，for循环枚举

class Solution {  
public:  
 int minPathSum(vector<vector<int>>& grid) {  
 int m = grid.size(), n = grid[0].size();  
 for(int i = 1; i < m; ++i)grid[i][0] += grid[i-1][0];  
 for(int i = 1; i < n; ++i)grid[0][i] += grid[0][i-1];  
 for(int i = 1; i < m; ++i){  
 for(int j = 1; j < n; ++j){  
 grid[i][j] += min(grid[i-1][j], grid[i][j-1]);  
 }  
 }  
 return grid[m-1][n-1];  
 }  
};