给定一个包含非负整数的 *m* x *n* 网格，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。

示例:

输入：

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

输出： 7

解释： 因为路径 1→3→1→1→1 的总和最小。

思路，该题只要路径，那么即为典型的动态规划题，第a[i][j]个格子最短的距离等于a[i][j-1],
a[i-1][j]间的最短距离加自己的权值，动态规划到a[m-1][n-1]即可.

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