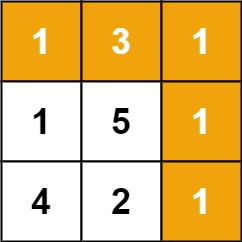
Given a m x n grid filled with non-negative numbers, find a path from top left to bottom right, which minimizes the sum of all numbers along its path.

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

**Example 1:**



**Input:** grid = [[1,3,1],[1,5,1],[4,2,1]]

**Output:** 7

**Explanation:** Because the path 1 → 3 → 1 → 1 → 1 minimizes the sum.

**Example 2:**

**Input:** grid = [[1,2,3],[4,5,6]]

**Output:** 12

**Solution:**

class Solution {

public int minPathSum(int[][] grid) {

if(grid == null || grid.length ==0)

return 0;

int[][] dp = new int[grid.length][grid[0].length];

for(int i=0; i< dp.length; i++){

for(int j=0; j<dp[i].length; j++){

dp[i][j] += grid[i][j];

if(i>0 && j>0){

dp[i][j] += Math.min(dp[i-1][j], dp[i][j-1]);

}

else if(i>0)

dp[i][j] += dp[i-1][j];

else if(j>0)

dp[i][j] += dp[i][j-1];

}

}

return dp[dp.length -1][dp[0].length -1];

}

}