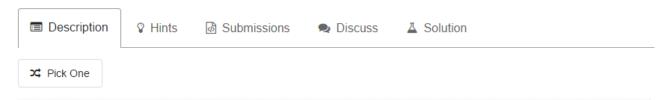# 64. Minimum Path Sum

⤨ Pick One

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.
```

```java
public class L64 {
/*
 *  这道题目是先处理最左边和最上边的两条边，因为只有一条路，
 *  接下来每一点的值都等于它上边和左边的较小值加上该点的值。
 *  即为到达该点的最短路径
 */
    public int minPathSum(int[][] grid) {
        int m = grid.length;
        int n = grid[0].length;

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

        return grid[m-1][n-1];
    }

}
```