

Minimum Path Sum

Question: Given a $m \times n$ 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.

Solutions:

class Solution:

@param grid, a list of lists of integers

@return an integer

def minPathSum(self, grid):

if len(grid)==0 or len(grid[0])==0:

return 0

for row in range(0, len(grid)):

for col in range(0, len(grid[0])):

if row>0 and col>0:

grid[row][col] += min(grid[row-1][col],grid[row][col-1])

elif row>0:

grid[row][col] += grid[row-1][col]

elif col>0:

grid[row][col] += grid[row][col-1]

return grid[len(grid)-1][len(grid[0])-1]

grid = [[1,2,3],

[4,5,6],

[7,8,9],

[10,11,12]]

Solution().minPathSum(grid)