# Grid

Little A has a grid of size $n \times m$ with a number written in each cell. For ease of description, let the cell in the top left corner be $(1, 1)$ and the cell in the bottom right corner $(n, m)$.

Little A can enter any of the cells in the bottommost (i.e., $n$th) row and play the game according to the following rules.

1. Let the first time he enters row $i$ be $(i, r_i)$. If he is at cell $(i, r_i)$, then he can only move left or up. Otherwise he can move left, right or up.

2. He cannot leave the grid, unless he is on the first row; leaving the grid from a cell in the first row ends the game.

Define the score of a game as the sum of the numbers on all the squares through which Little A passes. Little A would like you to help him find the lowest score he can get.

Input: the first lines contains $n, m$. The next $n$ lines contains $m$ integers, the values of the $i$th row.

### Subtask 1: $a_{i,j} \leq 0$

For 3 points. The answer is the sum of all the $a_{i,j}$.

### Subtask 2: $n, m \leq 5$

Brute force.

### Subtask 3: $n = 2$

We classify solutions by the rightmost selected cell of row 1. Fixing the index $i$ of rightmost selected cell, the solution consists of some subarray of the first row ending at $i$, some subarray of the second row ending at $i$, and some subarray of the second row begining at $i$; all these can be selected independently in $O(m)$ time by precomputing the prefix sum. Hence the total time is $O(m^2)$.

### Subtask 4: $n, m \leq 90$

Let $f_{i,l,r}$ be the minimum sum of the first $i$ rows if, at the last row, we select $[l, r]$. We have the recurrence

$$f_{i,l,r} = \left( \min_{l \leq x \leq y \leq r} f_{i-1,x,y} \right) + \sum_{j \in [l,r]} a_{ij}$$

Time complexity: from now on assume $m = n$. There are $O(n^3)$ DP states (because there are 3 variables $i, l, r$) and each state depends on $O(n^2)$ other states (because of the enumeration over $x$ and $y$). Hence this is $O(n^5)$. To reduce it, we can precompute $f'_{i,r} = \min_l f_{i,l,r}$.

## Subtask 5: $n, m \leq 400$

For every row our DP computes the answer if we select $l, r$ in the $i$th row. In subtask 4, after finishing the DP for a row, we precompute the optimum selection for fixed $r$, and use that to speed up the computation of the $i+1$st row. For this subtask, we can in addition precompute the optimum selection where $r \in [x, y]$ for each $x, y$. The preprocessing time per row increases to $O(n^2)$, but now the DP recurrence for the next row can be evaluated in $O(1)$. The total time complexity is $O(n^2)$.

## Subtask 6: $n, m \leq 10^3$

Use the recurrence $f_{i,j} = min(f_{i,r-1} + a_{i,j}, f_{i-1,r} + v_{i,j})$ where $v_{i,j}$ is the smallest subarray of line $i$ ending at $j$.