

## Kick Start 2017 - Round G

# Matrix Cutting

### Problem

Prof Shekhu has a matrix of **N** rows and **M** columns where rows are numbered from 0 to **N-1** from top to bottom, and columns are numbered from 0 to **M-1** from left to right. Each cell in the matrix contains a positive integer.

He wants to cut this matrix into **N \* M** submatrices (each of size  $1 * 1$ ) by making horizontal and vertical cuts. A cut can be made only on the boundary between two rows or two columns.

Prof Shekhu invites his best student Akki for this job and makes an interesting proposition. Every time Akki makes a cut in a submatrix, before he makes the cut, he is awarded a number of coins equal to the minimum value in that submatrix. Note that with every cut, the total number of submatrices increases. Also, cuts in any two different submatrices are independent and likewise, Akki is awarded independently for the cuts in different submatrices.

Now, Akki has various ways in which he can make the cuts. Can you help him by maximizing the total number of coins he can gain?

### Input

The first line of the input contains an integer **T**, the number of test cases. **T** test cases follow. The first line of each test case contains two integers **N** and **M**, as described above.

1. Next, there are **N** lines of **M** positive integers each; these describe the matrix.

### Output

For each test case, output one line containing `Case #x: y`, where **x** is the test case number (starting from 1) and **y** is the maximum possible number of coins that Akki can be awarded, if he makes the cuts in optimal order.

### Limits

$1 \leq T \leq 100$ .

Memory limit: 1GB.

$1 \leq \text{each value in the matrix} \leq 10^5$ .

### Small dataset (Test set 1 - Visible)

Time limit: 40 seconds.

**N** = 1.

$1 \leq M \leq 10$ .

### Large dataset (Test set 2 - Hidden)

Time limit: 120 seconds.

$1 \leq N \leq 40$ .

$1 \leq M \leq 40$ .

## Sample

### Sample Input

```
3
2 2
1 2
3 4
2 3
1 2 1
2 3 2
1 2
1 2
```

### Sample Output

```
Case #1: 5
Case #2: 7
Case #3: 1
```

In Sample Case #1, there are two possible ways in which Akki can make the cuts.

1. Suppose that Akki first cuts the matrix horizontally. He is awarded the minimum value in the matrix: 1. Then he has to make vertical cuts in the two submatrices ([1, 2] and [3, 4]), for which he gets 1 and 3 coins, respectively.
2. Suppose that Akki first cuts the matrix vertically. He is awarded the minimum value in the matrix: 1. Then he has to make horizontal cuts in the two submatrices (which have the transposes [1, 3] and [2, 4]), for which he gets 1 and 2 coins, respectively.

The first strategy is better, and the answer is 5.

In Sample Case #2, Akki can be awarded at most 7 coins. One of the optimal ways is to first make the only horizontal cut to earn 1 coin. Then, in the upper submatrix [1, 2, 1], Akki can first make the cut immediately to the right of first column and then the cut immediately to the right of second column to earn a total of 2 coins. Similarly, in the lower submatrix [2, 3, 2], Akki can first make the cut immediately to the right of second column and then the cut immediately to the right of first column to earn a total of 4 coins.

In Sample Case #3, there is only one cut to be made.