Maximum Matrix

Neo likes matrices. Neo's favourite type of *matrix* is a grid with R rows and C columns, where each cell in the matrix contains a positive integer.

Neo assigns a score (A, B) to each matrix:

- A is the number of *ascending* rows. A row is ascending if the values in this row are ascending when read from left to right. Specifically, if the values in the row are $v_1, v_2, \ldots, v_C$ from left to right, then the row is ascending if $v_1 \leq v_2 \leq \ldots \leq v_C$.
- B is the number of *constant* columns. A column is constant if the values in this column are all the same.

2	3	3	6	8
2	3	1	6	8
1	3	6	6	8

Figure 1: An example matrix with $R = 3$ rows and $C = 5$ columns. There are 2 ascending rows (the first and third) and 3 constant columns (the second, fourth, and fifth). Neo's score for this matrix is $(2, 3)$.

A matrix is *better* than another matrix if it has a lexicographically higher score. In particular, assume that you have one matrix with a score (A, B) and another matrix with a score (A', B'). The first matrix is better if one of the following conditions holds:

- $A > A'$, or
- $A = A'$ and $B > B'$.

For example,

- A matrix with score $(5, 3)$ is **better** than a matrix with score $(4, 4)$.
- A matrix with score $(5, 3)$ is **better** than a matrix with score $(5, 2)$.
- A matrix with score $(5, 3)$ is **not better** than a matrix with score $(5, 4)$.
- A matrix with score $(5, 3)$ is **not better** than a matrix with score $(6, 1)$.

You have found a matrix with some missing values. To impress Neo, you want to fill in the missing values with **positive integers** in a way that creates the best possible matrix. What is the score of the best matrix you can create?

Subtasks and Constraints

For all subtasks:

- $1 \le R \le 250\,000$ and $1 \le C \le 250\,000$.
- $R \times C \le 1\,000\,000$.
- All non-missing values in the matrix are positive integers from 1 to $1\,000\,000$, inclusive.

Additional constraints for each subtask are given below.

Subtask	Points	Additional constraints
1	7	$R = 1$.
2	18	The answer has $A = R$.
3	10	$R \le 10$, $C \le 10$, and every column has at least one value that is not missing.
4	8	$R \le 10$ and $C \le 10$.
5	17	$R \le 100$, $C \le 100$, and every column has at least one value that is not missing.
6	11	$R \le 100$ and $C \le 100$.
7	14	$R \le 5\,000$ and $C \le 5\,000$.
8	15	No additional constraints.

Input

- The first line of input contains the integers R and C.
- The next R lines of input each contain C integers, describing the Matrix. Each value in the matrix is either a positive integer or zero, where zero represents a missing value.

Output

Output two integers A and B on a single line, representing the score (A, B) of the best matrix that can be created.

Sample Input 1

```
3 5
2 3 3 6 0
0 3 1 6 8
1 3 6 0 8
```

Sample Output 1

```
2 3
```

Sample Input 2

```
2 3
1 0 2
3 0 4
```

Sample Output 2

```
2 0
```

Sample Input 3

```
2 4
2 4 0 1
2 0 3 1
```

Sample Output 3

```
0 4
```

Explanation

The first sample case has three missing values. One optimal way to fill in the missing values is to create the matrix shown in Figure 1, with $A = 2$ ascending rows and $B = 3$ constant columns.

The second sample case can be filled in as follows, with $A = 2$ ascending rows and $B = 0$ constant columns:

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

The third sample case can be filled in as follows, with $A = 0$ ascending rows and $B = 4$ constant columns:

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