Train stations

Description

Along a unidirectional railway (i.e. all the trains run in same direction), there are n train stations numbered $1 \ldots n$. All the stations have their own rank, the lowest of which is ranked 1.

Currently, there are some trains running on this railway. All of the trains should fulfill such a requirement: If this train stops at station x, it should also stop at all the stations between start station and destination station (these two included) that have a rank not lower than station x's rank.

For example, the table below shows a running schedule of 5 trains. In the table, D, S, F stand for departure station, stop station, final station respectively. Train 1, 2, 3 and 4 all fulfill the requirement. However, train 5 stops at station 3 (rank 2) but doesn't stop at station 6 (also rank 2), so it doesn't fulfill the requirement.

The number of train station	1		2		3		4		5		6		7		8
The level of train station	3		1		2		1		3		2		1		3
Train 1	D	→	→	→	S	→	→	→	S	→	F				
Train 2					D	→	→	→	S	→	F				
Train 3	D	→	→	→	→	→	→	→	S	→	→	→	→	→	F
Train 4							D	→	S	→	S	→	S	→	F
Train 5					D	→	→	→	S	→	→	→	→	→	F

Now provide the schedule of m trains, all of which fulfill the requirement. Please calculate at least how many different ranks can be assigned to these n stations.

Input

The first line contains two integers n, m, separated by a space.

In the following m lines, the i-th line $(1 \leq i \leq m)$ starts with one integer s_i $(2 \leq s_i \leq n)$ denoting that the i-th train has s_i stations to stop. The following s_i integers denoting the numbers of all the stations the train would stop. They are in increasing order and separated by spaces.

It is guarantees that all the trains fulfill the requirement.

Output

One integer indicating at least how many different ranks can be assigned.

Sample Input/Output

Input 1

```
9 2
4 1 3 5 6
3 3 5 6
```

Output 1

```
2
```

Input 2

```
9 3
4 1 3 5 6
3 3 5 6
3 1 5 9
```

Output 2

```
3
```

Constraints

$1 \leq n, m \leq 1000$.

Hint

The $O(n^2m)$ algorithm can pass through all test cases. In fact, the standard programme works in $O(nm)$ time complexity.