Problem A. Greedy Bipartite Matching

Input file:	*standard input*
Output file:	*standard output*
Time limit:	10 seconds
Memory limit:	1024 mebibytes

Consider a bipartite weighted graph with $2n$ vertices: n in the left part and n in the right part. The vertices in each part are numbered from 1 to n. A matching is called *greedy* if it has the maximal number of edges of weight 1 among all matchings, the maximal number of edges of weight 2 among all matchings that maximize the number of edges of weight 1, etc.

Your task is to find the size (number of edges) of greedy matching in a dynamically growing graph.

Input

The first line of the input contains two non-negative integers n and q ($n \le 10^5$, $q \le 10^3$): the number of vertices in each part and the number of different weights of the edges.

Then, the input consists of q blocks. The i-th block starts with a non-negative integer m_i: the number of edges of weight i. Each of the next m_i lines contains two integers x and y ($1 \le x, y \le n$), which add an edge between vertex x of the left part and vertex y of the right part. It is guaranteed that $\sum_i m_i \le 2 \cdot 10^5$.

Note that there may be multiple edges between two vertices.

Output

You have to output q integers on a single line: answers for the problem for weights at most 1, weights at most 2, ..., weights at most q.

Example

standard input	*standard output*
3 4 2 1 1 1 2 2 1 1 2 2 2 1 3 3 2 1 3 3	1 2 2 3