Bipartite Graph Matching Problem

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2.5 seconds
Memory limit:	1024 megabytes

You are given a bipartite graph. The left part contains n_1 vertices numbered from 1 to n_1, and the right part contains n_2 vertices numbered from 1 to n_2. There are m edges, each connecting one vertex on the left to one vertex on the right.

For every interval $[l, r]$ on the right side (where $1 \le l \le r \le n_2$), consider the subgraph obtained by:

- keeping all left vertices $1, 2, \ldots, n_1$;
- keeping only the right vertices with indices in $[l, r]$;
- keeping only edges whose endpoints are both kept.

In this subgraph, compute the size of a maximum matching (the maximum number of pairwise disjoint edges). Let this value be denoted by $f(l, r)$.

You need to compute the following value:

$$\sum_{1 \le l \le r \le n_2} f(l, r) \cdot l \cdot r \cdot ((l \oplus r) + 1),$$

where $\oplus$ denotes the bitwise XOR operation.

Output the value of this sum modulo 998244353.

Input

The input contains multiple test cases.

The first line contains a single integer T ($1 \le T \le 2000$), the number of test cases.

For each test case, the first line contains three integers $n_1, n_2, m (1 \le n_1, n_2 \le 5000, 0 \le m \le 10^4)$, representing the number of vertices on the left, the number of vertices on the right, and the number of edges.

Each of the next m lines contains two integers $u, v (1 \le u \le n_1, 1 \le v \le n_2)$, representing there is an edge between left vertex u and right vertex v.

It is guaranteed that there's no multiple edges, the sum of n_1 and the sum of n_2 does not exceed 5000.

Output

For each test case, output a single line containing one integer — the required sum modulo 998244353.

Example

standard input	standard output
4 3 3 5 1 1 2 1 2 2 2 3 3 3 3 4 6 1 2 1 4 2 3 2 1 3 4 3 3 2 2 1 1 2 4 3 5 1 3 2 1 3 3 4 2 4 1	81 529 12 81