Problem K. King of Swapping

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

You have a device for working with permutations $p_1, p_2, \ldots, p_n$ of integers from 1 to n. It can perform m operations. i-th operation is described by two integers a_i, b_i ($1 \le a_i, b_i \le n, a_i \ne b_i$). If you apply it, it will do the following: if $p_{a_i} > p_{b_i}$, then the device will swap p_{a_i}, p_{b_i}. Otherwise, it will do nothing.

You can apply these operations any number of times, in any order (you can use one operation more than once).

You are interested in whether you can use this device to get every permutation from every other permutation. In other words, determine whether for every two permutations $p_1, p_2, \ldots, p_n$ and $q_1, q_2, \ldots, q_n$ of the integers from 1 to n, there exists a sequence of operations that can be applied to p to obtain q.

Recall that the permutation of integers from 1 to n is an array of length n containing each integer from 1 to n exactly once.

Input

The first line contains a single integer t ($1 \le t \le 10^4$) — the number of test cases. The description of test cases follows.

The first line of each test case contains two integers n, m ($1 \le n \le 2 \cdot 10^5, 0 \le m \le 2 \cdot 10^5$) —the length of permutations your device can handle and the number of operations it can perform.

The i-th of the m following lines contains two integers a_i, b_i ($1 \le a_i, b_i \le n, a_i \ne b_i$) describing the i-th operation: if $p_{a_i} > p_{b_i}$, the device can swap the elements p_{a_i}, p_{b_i} of the permutation.

It is guaranteed that all pairs (a_i, b_i) are pairwise distinct, but note that pairs (x, y) and (y, x) may occur simultaneously for some x, y.

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$, and the sum of m over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, print `YES` if you can obtain any permutation from any other permutation with this device, and `NO` otherwise.

You can print `YES` and `NO` in any case (e.g. the strings `yEs`, `yes`, `Yes` will be taken as a positive answer).

Example

standard input	standard output
5 2 1 1 2 3 4 1 2 2 1 1 3 3 1 5 4 1 2 2 3 3 4 4 5 5 6 3 5 5 1 1 2 4 3 2 4 4 1 4 6 3 1 2 3 2 1 3 2 1 2 1 3	NO YES NO YES NO

Note

In the first example, we can swap p_1 and p_2 if $p_1 > p_2$. Thus, we can get the permutation $(2, 1)$ from the permutation $(1, 2)$, but we cannot get the permutation $(1, 2)$ from the permutation $(2, 1)$, so the answer is `NO`.

In the second example, we can swap p_1, p_2 regardless of which one is larger (because we have both swap operations: with $a_i = 1, b_i = 2$ and with $a_i = 2, b_i = 1$). Also, we can swap p_1, p_3 regardless of which of them is greater. It is easy to show that in this case we can get from any permutation any other permutation, so the answer is `YES`.