Problem H. Consistency

Input file:	standard input
Output file:	standard output
Time limit:	1 seconds

In mathematics a theory T is said to be consistent if it does not lead to a contradiction, meaning that there is no sentence x such that x and its negation are both true. In 1931 Kurt Gödel stated that a theory cannot prove its consistency. Here, we would like to define a *petite version* of a provable consistency .

Let T be a finite set of logical sentences. Each two members x and y of T are either equivalent (we denote $x \Leftrightarrow y$) or not. As you may be aware, equivalence relations are symmetric (if $x \Leftrightarrow y$ then $y \Leftrightarrow x$ for all pairs x and y of T), reflexive ($x \Leftrightarrow x$ for all members x of T) and transitive (if x, y and z are members of T such that x⇔y and y⇔z then x⇔z).

If there exist three members of T such that the transitivity of the equivalence relation ⇔ does not apply, then T is said to be inconsistent, otherwise T is consistent. We suppose that symmetricity and reflexivity properties are always met.

Here is an example : suppose that T consists of four members identified by numbers from 1 to 4 such that $1 \Leftrightarrow 3$ and $3 \Leftrightarrow 4$, all the remaining pairs of distinct members are not equivalent. In this case, T is inconsistent, because the transitivity does not apply to 1, 3 and 4 ($1 \Leftrightarrow 3$ and $3 \Leftrightarrow 4$ but 1 and 4 are not equivalent). Not that 2 is not equivalent to any of the other members which is totally fine.

Write a program that determines if T is consistent or not.

Input

The first line of the input contains two integer numbers n and m $(\leq n, m \leq 10^5)$ where n is the number of the sentences in T, and m is the number of distinct pairs that are equivalent. Sentences of T are identified by integers from 1 to n.
m lines follow, each of them contains two distinct integers x and y $(1 \leq x, y \leq n \;\; x \neq y)$ indicating that x and y are equivalent. Each pair of distinct sentences that are equivalent will appear exactly once in the input.

Output

If T is consistent print “YES”, if T is inconsistent print “NO”.

Example

Standard input	Standard output
4 2 1 3 3 4	NO
4 3 1 3 3 4 1 4	YES
3 3 1 2 2 3 3 1	YES