

Problem B

Cactus

Input: standard input
Output: standard output
Time Limit: 5 seconds

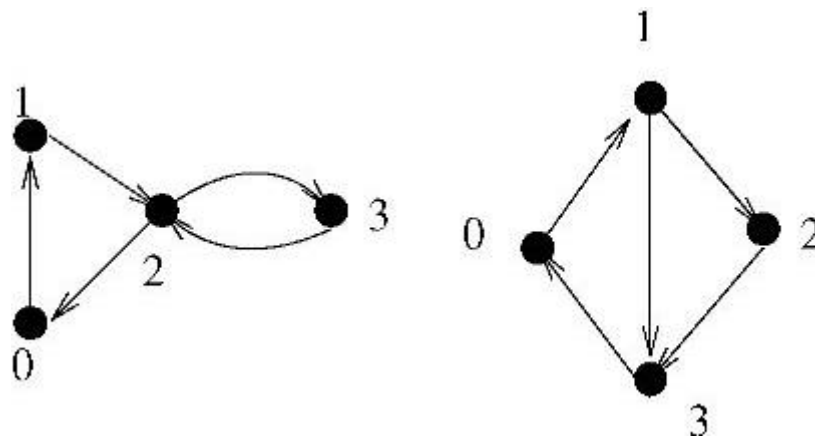
A directed graph is a set V of *vertices* and a set of $E \subseteq \{V \times V\}$ *edges*. An edge (u,v) is said to be directed from u to v (the edge (v,u) has the opposite direction). A directed cycle in a directed graph is a sequence of edges

$(u_1, v_1), (u_2, v_2), \dots, (u_k, v_k)$

such that $u_{i+1} = v_i$ for $i = 1, \dots, k-1$, and $u_1 = v_k$. The directed cycle is *simple* if $u_i \neq u_j$ whenever $i \neq j$ (i.e., if it does not pass through a vertex twice).



In a *strongly connected* directed graph, there is for every pair u, v of vertices some directed cycle (not necessarily simple) that visits both u and v .



A directed graph is a *cactus* if and only if it is strongly connected and each edge is part of exactly one directed simple cycle. The first graph is a cactus, but the second one is not since for instance the edge $(0,1)$ is in two simple cycles.

The reason for the name is that a "cactus" consists of several simple cycles connected to each other in a tree-like fashion, making it look somewhat like a cactus.

Problem

Write a program that given a directed graph determines if it is a cactus or not. The graph can have several thousand vertices.

Input

The first line contains an integer, which is the number of test cases (less than 20). Each test case starts a line with an integer $n \geq 0$ followed by line with an integer $m \geq 0$ giving the number of vertices (n) and edges (m) in a graph (at most 10,000 of each). The vertices are

numbered **0** through **n-1**. The following **m** lines describe the edges as pairs of numbers **u v** denoting an edge directed from **u** to **v**. There will never be more than one edge from **u** to **v** for any pair of vertices **u** and **v**. There are no loops, i.e., no edges from a vertex to itself.

Output

For each test case output a single line with a single string. Output **"YES"** if the graph is a cactus, and output **"NO"** if it is not.

Sample Input

```
2
4
5
0 1
1 2
2 0
2 3
3 2
4
5
0 1
1 2
2 3
3 0
1 3
```

Sample Output

```
YES
NO
```

Problemsetter: M. Goldmann