



Problem F. Face Recognition Algorithm

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 256 mebibytes

I have drawn n dots on a plane and m straight segments connecting these dots. No segment goes through dots other than its endpoints, and no two segments intersect in any point other than their common endpoint. Also, if you start in one dot and move only by segments, you can go to any other dot. All n dots are in distinct positions.

Yes, that's a planar embedding of some connected graph with n vertices and m edges. Your task is to check if each face of this graph, including the outer face, is a triangle. Face is a triangle if and only if it is bounded by exactly 3 edges. If face is on both sides of some edge, this edge is counted twice.

Strive for excellence! Allow yourself nothing less than the best possible complexity for your *algorithm*.

Input

The first line contains two integers n and m ($3 \leq n \leq 10^5$, $n - 1 \leq m \leq 3 \cdot 10^5$) — the number of dots and the number of segments, respectively.

The next n lines contain coordinates of dots. All the coordinates are not greater than 10^9 by absolute value. All n dots are in distinct positions.

Each of the next m lines contains two integers u and v ($1 \leq u, v \leq n$, $u \neq v$) — the ids of dots connected by a segment. It is guaranteed that there are no two segments connecting the same pair of dots.

It is guaranteed that the input describes a valid planar embedding of a connected graph with all edges drawn as straight segments.

Output

If each face of the given graph, including its outer face, is a triangle, print “YES”, otherwise print “NO”.

Examples

standard input	standard output
3 3 0 0 1 0 0 1 1 2 2 3 3 1	YES
4 4 0 0 3 0 0 3 1 1 1 2 2 3 3 1 1 4	NO
4 6 0 0 3 0 0 3 1 1 1 2 2 3 3 1 1 4 2 4 3 4	YES
4 5 0 0 2 0 1 1 0 2 1 2 1 3 1 4 2 3 3 4	NO
4 3 0 0 0 1 1 1 1 2 1 2 2 3 3 4	NO