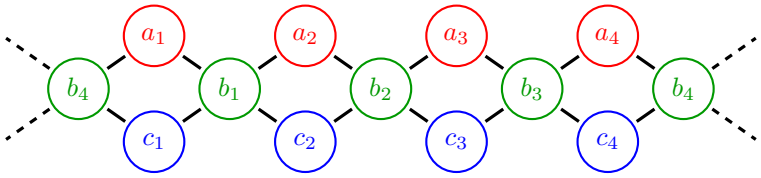
Problem A. Arrange Jewels In Necklace

Time limit:	1 second
Memory limit:	1024 mebibytes

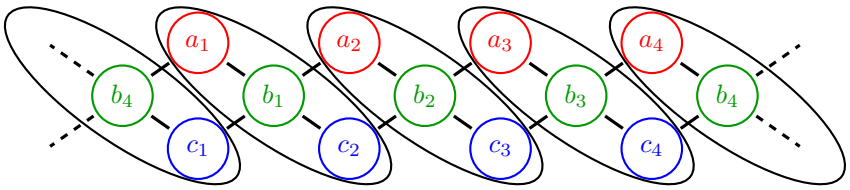
Boris wants to sell beautiful necklaces. He consulted the best designers and jewelers to invent the best necklace. Now he needs your help to understand if there exists a perfect necklace meeting his expectations.

A necklace that Boris wants to make consists of $3k$ jewels, where k is the length of the necklace. Any length $k \geq 2$ is allowed. They are placed in three rows, each having k jewels. All jewels have types (identified with integers). Let us define types of jewels from left to right in the first row as $a_1, a_2, \ldots, a_k$, in the second row as $b_1, b_2, \ldots, b_k$, and in the third row as $c_1, c_2, \ldots, c_k$.

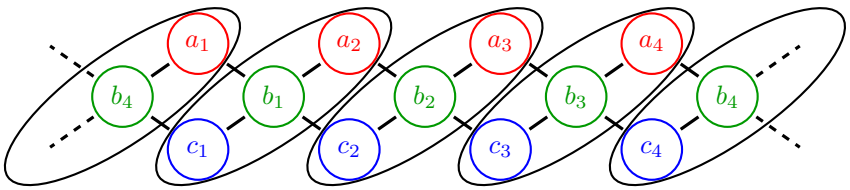


An example of a necklace for $k = 4$

A necklace has k backward columns and k forward columns. Each column contains 3 jewels. The order of jewels in each column is important. Backward columns are $(a_1, b_1, c_2), (a_2, b_2, c_3), \ldots, (a_i, b_i, c_{i+1}), \ldots, (a_k, b_k, c_1)$. Forward columns are $(a_2, b_1, c_1), (a_3, b_2, c_2), \ldots, (a_{i+1}, b_i, c_i), \ldots, (a_1, b_k, c_k)$.



Backward columns



Forward columns

Boris has n options $\{(x_i, y_i, z_i)\}_{i=1}^{n}$ of beautiful backward columns, and m options $\{(u_i, v_i, w_i)\}_{i=1}^{m}$ of beautiful forward columns. He says that a necklace is beautiful if all its columns are beautiful (each column is present in the corresponding beautiful columns set). Find if there exists any beautiful necklace with length ≥ 2.

Also, Boris guarantees that all y_i are different. Note that it is guaranteed only for the middle jewels of beautiful backward columns, so any other numbers may be equal.

Input

Each input consists of multiple test cases. The first line contains a single integer t ($1 \leq t \leq 10^5$) — the number of test cases. Test case descriptions follow.

The first line of each test case contains two integers n, m ($1 \leq n, m \leq 10^5$) — the number of beautiful backward columns and the number of beautiful forward columns.

Each of the next n lines contains three integers x_i, y_i, z_i ($1 \leq x_i, y_i, z_i \leq 10^9$), describing the next beautiful backward column.

Each of the next m lines contains three integers u_i, v_i, w_i ($1 \le u_i, v_i, w_i \le 10^9$), describing the next beautiful forward column.

It is guaranteed that in each test case all y_i are different.

It is guaranteed that the sum of n for all test cases and the sum of m for all test cases does not exceed 10^5.

Output

For each test case, print a single word "YES" if a beautiful necklace with length ≥ 2 exists, and "NO" otherwise.

You can print each letter in any case.

Example

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

Note

An example of a beautiful necklace (with length 4) for the first test case:

