DMOPC '22 Contest 1 P6 - Linear Swap

Time limit: 2.0s **Memory limit:** 256M

You are given an array A of N integers. There is a list of M operations that you must apply to the array in some order. The i-th operation has 3 parameters a_i, b_i, and s_i, and it can be applied by calling the following function:

```
bool apply(int a, int b, int s) {
    int C = a;
    for (int i = s; i <= N; i++) {
        if (C > A[i]) swap(C,A[i]);
    }
    return C <= b;
}
```

The operation is successfully applied if and only if the function returns `true`. Your task is to determine whether there is some ordering of the operations such that all of the operations can be applied successfully. To ensure the integrity of your solution, there may be up to T test cases.

Constraints

$1 \leq T \leq 5 \times 10^5$

$1 \leq N, M \leq 5 \times 10^5$

$1 \leq A_i, a_i, b_i \leq 10^9$

$1 \leq s_i \leq N$

The sum of N over all test cases does not exceed 5×10^5.

The sum of M over all test cases does not exceed 5×10^5.

Subtask 1 [10%]

The sum of N over all test cases does not exceed 15.

The sum of M over all test cases does not exceed 15.

Subtask 2 [20%]

The sum of N over all test cases does not exceed 5×10^3.

The sum of M over all test cases does not exceed 5×10^3.

Subtask 3 [70%]

No additional constraints.

Input Specification

The first line contains an integer T, the number of test cases. The remaining lines describe the test cases.

The first line of each test case contains 2 integers N and M.

The second line of each test case contains N integers $A_1, A_2, \ldots, A_N$.

The remaining M lines of each test case each contain 3 integers a_i, b_i, and s_i, describing the operations of the test case.

Output Specification

For each test case, output one line containing `YES` if there is an ordering of the operations such that all of the operations can be applied successfully, or `NO` otherwise.

Sample Input

```
1
5 3
3 1 4 1 5
2 3 5
6 1 1
3 2 3
```

Sample Output

```
YES
```

Explanation for Sample

One possible solution is as follows:

1. Apply the third operation. This transforms A into $[3, 1, 4, 3, 5]$. Since $C = 1$ at the end of the process and $1 \leq 2$, this operation was applied successfully.
2. Apply the first operation. This transforms A into $[3, 1, 4, 3, 5]$. Since $C = 2$ at the end of the process and $2 \leq 3$, this operation was applied successfully.
3. Apply the second operation. This transforms A into $[6, 3, 4, 3, 5]$. Since $C = 1$ at the end of the process and $1 \leq 1$, this operation was applied successfully.