E. Wolf

time limit per test: 4 seconds
memory limit per test: 256 megabytes

Wolf has found n sheep with tastiness values $p_1, p_2, \ldots, p_n$ where p is a permutation*. Wolf wants to perform binary search on p to find the sheep with tastiness of k, but p may not necessarily be sorted. The success of binary search on the range $[l, r]$ for k is represented as $f(l, r, k)$, which is defined as follows:

If $l > r$, then $f(l, r, k)$ fails. Otherwise, let $m = \lfloor \frac{l+r}{2} \rfloor$, and:

- If $p_m = k$, then $f(l, r, k)$ is **successful**,
- If $p_m < k$, then $f(l, r, k) = f(m+1, r, k)$,
- If $p_m > k$, then $f(l, r, k) = f(l, m-1, k)$.

Cow the Nerd decides to help Wolf out. Cow the Nerd is given q queries, each consisting of three integers l, r, and k. Before the search begins, Cow the Nerd may choose a non-negative integer d, and d indices $1 \le i_1 < i_2 < \ldots < i_d \le n$ where $p_{i_j} \neq k$ over all $1 \le j \le d$. Then, he may re-order the elements $p_{i_1}, p_{i_2}, \ldots, p_{i_d}$ however he likes.

For each query, output the **minimum** integer d that Cow the Nerd must choose so that $f(l, r, k)$ can be **successful**, or report that it is impossible. Note that the queries are independent and the reordering is not actually performed.

*A permutation of length n is an array that contains every integer from 1 to n exactly once.

Input

The first line of the input contains a single integer t $(1 \le t \le 10^4)$ — the number of test cases.

The first line of each test contains two integers n and q $(1 \le n, q \le 2 \cdot 10^5)$ — the length of p and the number of queries respectively.

The second line contains n integers $p_1, p_2, \ldots, p_n$ — the tastiness of the i-th sheep. It is guaranteed that every integer from 1 to n appears exactly once in p.

The following q lines contain three integers l, r, and k $(1 \le l \le r \le n, 1 \le k \le n)$ — the range that the binary search will be performed on and the integer being searched for each query.

It is guaranteed that the sum of n and the sum of q over all cases do not exceed $2 \cdot 10^5$.

Output

For each query, output the minimum integer d that Cow the Nerd must choose so that $f(l, r, k)$ is successful on a new line. If it is impossible, output -1.

Example

input Copy

```
8
5 3
1 2 3 4 5
1 5 4
1 3 4
3 4 4
7 4
3 1 5 2 7 6 4
3 4 2
2 3 5
1 5 6
1 7 3
2 1
2 1
1 2 1
1 1
1
1 1 1
7 1
3 4 2 5 7 1 6
1 7 1
```

Codeforces Round 1020 (Div. 3)
Finished
Practice

→ Virtual participation

Virtual contest is a way to take part in past contest, as close as possible to participation on time. It is supported only ICPC mode for virtual contests. If you've seen these problems, a virtual contest is not for you - solve these problems in the archive. If you just want to solve some problem from a contest, a virtual contest is not for you - solve this problem in the archive. Never use someone else's code, read the tutorials or communicate with other person during a virtual contest.

Start virtual contest

→ Clone Contest to Mashup

You can clone this contest to a mashup.

Clone Contest

→ Submit?

Language: GNU G++23 14.2 (64 bit, ms

Choose file: Choose File No file chosen

Submit

→ Last submissions

Submission	Time	Verdict
317335454	Apr/26/2025 15:46	Accepted

→ Problem tags

binary search greedy math

No tag edit access

→ Contest materials

- Announcement (en)
- Tutorial #1 (en)
- Video Tutorial (en)

```
16 1
16 10 12 6 13 9 14 3 8 11 15 2 7 1 5 4
1 16 4
16 1
14 1 3 15 4 5 6 16 7 8 9 10 11 12 13 2
1 16 14
13 1
12 13 10 9 8 4 11 5 7 6 2 1 3
1 13 2
```

output

```
0 -1 0
2 0 -1 4
-1
0
-1
-1
-1
-1
```

Note

In the first example, second query: Since 4 does not exist in the first three elements, it is impossible to find it when searching for it in that range.

In the second example, on the first query, you may choose the indices 2, 3, and swap them so $p = [3, 5, 1, 2, 7, 6, 4]$. Then, $f(3, 4, 2)$ will work as follows:

1. Let $m = \lfloor \frac{3+4}{2} \rfloor = 3$. Because $p_3 = 1 < 2$, then $f(3, 4, 2) = f(4, 4, 2)$.
2. Let $m = \lfloor \frac{4+4}{2} \rfloor = 4$. Because $p_4 = 2 = k$, then $f(4, 4, 2)$ is successful. Therefore, $f(3, 4, 2)$ is also successful.

The total indices chosen were 2, so the final cost is 2, which can be shown that is minimum. Note that for this query we can't choose index 4, since $p_4 = k = 2$.

In the last query of the second example, we can choose indices $2, 3, 4, 5$ and re-arrange them so $p = [3, 5, 2, 7, 1, 6, 4]$. Then, $f(1, 7, 3)$ is successful.