Sphere online judge
(/)

PROBLEMS (/problems) ● STATUS (/status) RANKS (/ranks)

DISCUSS (http://spoj.com/forum) CONTESTS (/contests) ⚠ (/manageaccount/)

PROFILE

Problems (/problems) / classical (/problems/classical) / K-Query Online

My status (/status/KQUERYO,nadim_mahmud/) Status (/status/KQUERYO/) Ranking (/ranks/KQUERYO/)

KQUERYO - K-Query Online

no tags

Given a sequence of n numbers a_1, a_2, ..., a_n and a number of k-queries. A k-query is a triple (i, j, k) (1 ≤ i ≤ j ≤ n). For each k-query (i, j, k), you have to return the number of elements **greater than** k in the subsequence a_i, a_{i+1}, ..., a_j.

Input

- Line 1: n (1 ≤ n ≤ 30000).
- Line 2: n numbers a_1, a_2, ..., a_n ($1 \le a_i \le 10^9$).
- Line 3: q (1 ≤ q ≤ 200000), the number of k- queries.
- In the next q lines, each line contains 3 numbers a, b, c representing a k-query. You should do the following:
    - i = a xor last_ans
    - j = b xor last_ans
    - k = c xor last_ans

  After that 1 ≤ i ≤ j ≤ n, $1 \le k \le 10^9$ holds.
  Where last_ans = the answer to the last query (for the first query it's 0).

Output

For each k-query (i, j, k), print the number of elements greater than k in the subsequence a_i, a_{i+1}, ..., a_j in a single line.

Example

```
Input:
6
8 9 3 5 1 9
5
2 3 5
3 3 7
0 0 11
0 0 2
3 7 4

Output:
1
1
0
0
2
```

[**Edited by EB**]

There are invalid queries. Assume the following:

- if i < 1: i = 1
- if j > n: j = n
- if i > j: ans = 0

Submit solution! (/submit/KQUERYO/)

hide comments

< | Previous | 1 | 2 (/problems/KQUERYO/cstart=10) | 3 (/problems/KQUERYO/cstart=20) | Next (/problems/KQUERYO/cstart=10) | > (/problems/KQUERYO/cstart=30)

Erick (/users/ericksav22): 2018-08-28 07:14:00
Finally AC!
Tip: Don't use the third if putted by EB (i > j: ans = 0) it causes WA!!

tisparta (/users/tisparta): 2018-08-09 05:43:20
Also solvable with Wavelet tree e.e

amulyagaur (/users/amulyagaur): 2018-03-14 07:12:34
same concept can be applied here as well:
https://www.codechef.com/problems/PRMQ

madhur4127 (/users/madhur4127): 2018-02-26 14:50:18
O(sqrt(N)*log(N)) gives AC in 0.12s, how to reduce time other than using merge sort tree?

ayushgupta1997 (/users/ayushgupta1997): 2018-02-22 14:05:42
The test cases are weak sqrt decomp also passes...even i didn't use long long for a[] still passed :(

ramini1996 (/users/ramini1996): 2018-02-05 11:05:10
AC in ONE GO !!!

shiv2111 (/users/shiv2111): 2018-01-12 10:32:34
same version KQUERY has very strict TL, merge sort tree is not going to work there.

sherlock726 (/users/sherlock726): 2017-10-16 20:56:51
ac on first go
simple segment tree + vector for each node

sajib_only (/users/sajib_only): 2017-07-03 18:43:07
Don't know what was the problem with this problem. I thought I will get WA but got AC. have many questions.
#are the i and j 0 based or 1 based? (i got AC with assuming 1 based but i don't know what happens if we get 0 after XOR :3)
#got a WA when i tool 4*30000 + 10 as size of the Segment Tree Array. then when i made it 4*300000 + 10 , it passed smoothly with AC

prakhar10_10 (/users/prakhar10_10): 2017-06-28 06:53:04
The test data is still wrong I think.

Leave a Comment

Publish

Notes:
1. Don't post any source code here.
2. Please be careful, leave short comments only. Don't spam here.
3. For more discussion (hints, ideas, solutions) please visit our forum (/forum).
4. Authors of the problems are allowed to delete the post and use html code here (e.g. to provide some useful links).

Submit solution! (/submit/KQUERYO/)

Added by:	amirmd76 (/users /amirmd76)
Date:	2015-04-17

Time limit:	0.200s
Source limit:	50000B
Memory limit:	1536MB
Cluster:	Cube (Intel G860) (/clusters/)
Languages:	All

Evaluate this problem

Nobody has rated this problem yet, maybe you'll be the first?

Concept difficulty

easy normal hard extreme

Implementation difficulty

easy normal hard extreme

Recommend!

Own tags

#sqrt-decomp-2 # # # # # # # #

Tag name Add

About (/info) | Tutorial (/tutorials) | Tools (/tools) | Clusters (/clusters) | Credits (/credits) | API (/sphereengine) | Widgets (/sphereengine-widget)

Legal: Terms of Service (/legal-tos/) | Privacy Policy (/legal-pp/) | GDPR Info (/legal-gdpr/)

RSS (/rss/)