IOI Training Camp 2014 – Test 2, 2 May, 2014

Problem 1 Rank in Subarray

Today, we have a very interesting problem for you. Given an array A of N integers indexed from 1 to N, you need to perform following two types of queries :

- Change the value of $A[x]$ to k.
- Find the kth ranked element in the subarray $A[x..y]$ (x and y inclusive). An element is said to have kth rank if its position is k when the subarray is sorted in ascending order.

Can you perform these queries efficiently?

Input format

- The first line contains a single integer T, denoting the number of test cases. The first line of each test case contains an integer N denoting the size of the array. The next line contains N space separated integers where the ith integer represents $A[i]$ ($1 \leq i \leq N$).
- The next line contains an integer Q denoting the number of queries to be performed.
- Q lines follow with each line representing a query.

  Queries can be of two types:

  - `1 x k` – Update the xth element to k; i.e., set $A[x] = k$, $1 \leq x \leq N$.
  - `0 x y k` – Find the kth ranked element in the subarray $A[x..y]$, $1 \leq x \leq y \leq N$, $1 \leq k \leq y - x + 1$.

Output format

For every test case, output the result of each query of the second type on a new line.

Test data

For all subtasks, $1 \leq T \leq 5$, $1 \leq N \leq 10^4$, $1 \leq Q \leq 5 \times 10^4$, $1 \leq A[i] \leq 1000$.

- **Subtask 1 (20 marks)** Only queries of the second type will be present (i.e., queries of the form `0 x y k`)
- **Subtask 2 (30 marks)** Both types of queries will be present, but queries of the second type will always be of the form `0 1 N k`.
- **Subtask 3 (50 marks)** No additional constraints.

Sample input

```
2
2
1 2
2
0 1 2 1
0 1 2 2
4
4 3 2 1
4
0 1 1 1
1 1 1
0 1 1 1
0 1 4 3
```

Sample output

```
1
2
4
1
2
```

Explanation There are two test cases:

- The first test case contains only two elements which can be represented as $[1, 2]$. Two queries are to be processed :

  (i) The first query asks for smallest element in the range $A[1..2]$, which is 1.

  (ii) The second query asks for the 2nd smallest element in the range $A[1..2]$, which is 2.

- The second test case contains four elements which can be represented as [4, 3, 2, 1]. Four queries are to be processed :

  (iii) The first query asks for the smallest element in $A[1..1]$ which is 4.

  (iv) The second query updates the 1st element to 1. Hence the array now looks like $[1, 3, 2, 1]$.

  (v) The third query asks for the smallest element in $A[1..1]$, which is now 1.

  (vi) The last query asks for the 3rd smallest element in $A[1..4]$, which is 2.

Limits

- *Memory limit* : 256 MB
- *Time limit* : 2s