(Adv.) Competitive Programming

Submit until 14.06.2019 13:30, via the judge interface

HPI Hasso Plattner Institut
Digital Engineering · Universität Potsdam

Problem: noise-removal (1 second timelimit)

To make some extra money during your studies you work at a hardware company. Your task is write a new driver for an old sensor that the company stopped producing ages ago but is contractually obligated to support. All documentation has since been lost, so your only hope is reverse engineering the old driver and sensor data. Some painful hours later you gathered the following:

- The sensor only sends a signal when a specific event happens. For each event, it sends values that steadily increase (each value higher than the previous). This reverses at some point, so that the values steadily decrease.
- Due to a hardware design error the sensors data contains lots of noise. It might even send noise before the first and after the last value send during an event.
- The only interesting aspect of the values is how many of them (excluding noise) are send during an event. Due to the noise this number can only be estimated. It should only ever be overestimated, never the other way around.

Input The input begins with r ($0 < r \leq 50$), the number of recorded signal snippets to test. Each signal recording contains the values received for one event and is represented as follows: A first line contains n ($0 < n \leq 5000$), the number of values recorded. The second line contains all n values v_i ($0 \leq v_i < 2^{32}$) in the order they were received, separated by a space.

Output For each recording, output a line containing the best safe estimate you can make for the number of non-noise values send by the sensor.

Sample input

```
2
10
9 7 2 5 6 1 3 4 0 8
6
1 1 2 2 1 1
```

Sample output

```
5
3
```