Skill

BOI 2023 task proposal, Latvia

Statement

This is an interactive problem. There is a tournament with N players participating, numbered from 1 to N. The skill of the i-th player is a positive integer number a_i. The skills of all players are pairwise distinct.

Initially all of the skills are unknown to you, and your task is to determine for each player the value of their skill. To do that, you can ask queries; each query takes two indices i, j and returns

$$\min(a_i, a_j).$$

Since it is impossible to find the skills of all players using such queries, you need to find the skills of only $N - 2$ players. You can use at most 10^5 queries.

The interactor is non-adversarial, meaning that the skills a_1, a_2, …, a_N are fixed and are not changing during the execution on each test.

Interaction

The first line of the input contains an integer N ($3 \leq N \leq 99800$), the number of players. Then you can ask queries: to ask a query, you need to output a question mark `?` followed by two indices i, j ($1 \leq i, j \leq N$, $i \neq j$). The interactor will then write in the input a positive integer that is $\min(a_i, a_j)$.

To output the answer, first output an exclamation mark `!` followed by N numbers, where the i-th number should be either a_i or 0 in case you don't know the skill of the i-th player. There should be exactly two 0 among these N numbers. It can be proven that the answer is unique.

Sample

Input	Output
4	
	? 1 4
5	
	? 4 3
1	
	? 1 2
5	
	! 5 0 1 0

Subtask suggestions

1. $N \leq 30\,000$.
2. $N \leq 50\,000$.
3. $N \leq 96\,000$.
4. No additional constraints.

Comments

This is an interactive task not difficult for people who are aware of probabilistic approach. The limits on the subtasks should be chosen with care.

Author: Jevgēnijs Vihrovs, jevgeniy.vihrov@gmail.com, Latvian OI problem committee.

Spoiler

First, it is easy to see that the two skills that cannot be determined are the two largest skills. Let Q be the number of queries used by the solution.

Subtask 1. $Q \leq 3N$, deterministic solution.

Let's process all the indices one by one and keep the indices j and k of the two largest skills encountered so far. At first, $j = 1$ and $k = 2$. Then let i be the index we are processing. By asking 3 queries $\min(a_i, a_j)$, $\min(a_j, a_k)$ and $\min(a_i, a_k)$, we determine the position of the smallest of these three skills. The other two remain the largest skills encountered so far.

Subtask 2. $Q \leq 2N$, deterministic solution.

In the previous solution, we can notice that we need to ask only two more queries for each next number we process, as the minimum of the two largest known skills is already known from the previous queries.

Another solution: split all indices into $N/2$ pairs, and query each of those pairs. Let the maximum of the minimums of these pairs be m. Since all numbers are distinct, we have m greater than all other $N/2 - 1$ minimums. We also have for this pair $\{i, j\}$, both $a_i, a_j \geq m$. Then we make $N/2-1$ queries with $a_i \geq m$ and one of the positions in each pair $\{k, \ell\}$, say, k. If $\min(a_i, a_k) = \min(a_k, a_\ell)$, then we know that $a_k = \min(a_k, a_\ell)$. Otherwise we know that $a_\ell = \min(a_k, a_\ell)$. In this way we find $N/2$ numbers using N queries; then we proceed in the same way recursively with the unknown skills. The total number of queries is at most $N + N/2 + N/4 + \ldots < 2N$.

Subtask 3. $Q = N + O(\sqrt{N})$, probabilistic solution.

Choose random $t = 10\sqrt{N}$ elements and solve the problem for this set using the algorithm from Subtask 2. It can be proven that the two largest elements of these t with high probability are among the $\sqrt{N}$ largest. Let one of these positions be i, then we check all other $N - t$ elements with i. In this way we will find all of the $N - \sqrt{N}$ smallest elements. To determine the rest, we again solve it using the algorithm from Subtask 2. The total number of queries is

$$2t + (N - t) + 2\sqrt{N} = N + 12\sqrt{N}.$$

Now we estimate the probability that a_i is one of the $\sqrt{N}$ largest elements. For this, we will estimate the probability that the two largest elements of the $10\sqrt{N}$ chosen are at least $\sqrt{N}$-th largest.

The probability that a randomly chosen element among the $\sqrt{N}$ largest elements is $\frac{1}{\sqrt{N}}$. The probability that out of $5\sqrt{N}$ randomly chosen elements none of them is among the $\sqrt{N}$ largest is

$$\left(1 - \frac{1}{\sqrt{N}}\right)^{5\sqrt{N}} \approx \frac{1}{e^5}.$$

By the union bound, the probability that the second largest element is not among the $\sqrt{N}$ largest is at most

$$\frac{2}{e^5} < 1.4\%.$$

Note 1. by solving the subproblems recursively using the same algorithm, we can improve the number of queries even more, but the complexity is still $N + O(\sqrt{N})$, so it is not so interesting.

Note 2. The estimates here are not optimal, but since we shouldn't expect the contestants to optimize an asymptotically correct solution, the limit in this subtask is somewhat loose. Intuition is enough.

Subtask 4. $Q = N + O(\log N)$, probabilistic solution.

In this solution we process the elements in random order one by one. We keep the indices of the two largest elements i and j, and we know $\min(a_i, a_j)$. With two queries we can determine this for the first two elements.

When we examine a new element at position k, we query $\min(a_i, a_k)$. Then we have two cases:

(a) If $\min(a_i, a_k) = a_k$, then we know the value at k.

(b) If $\min(a_i, a_k) > a_k$, then we have to query $\min(a_j, a_k)$; then we know the value and the position of the smallest element of the three, and we know the positions of the two largest elements and their minimum. Thus, in this case we use two queries.

Now we calculate the complexity of this solution. It is

$$N + X,$$

where $X =$ the number of occurrences of case (b). Case (b) happens when a_k is one of the two largest elements among the numbers a_1, a_2, …, a_k. This probability is $2/k$.

Thus, the expectation that this event will occur at k is $2/k$. The total expected number of such occurrences then is

$$\frac{2}{3} + \frac{2}{4} + \ldots + \frac{2}{N} \approx 2 \ln N$$

by the harmonic number approximation. Intuitively, this is enough to try this solution, and it works.

Probability. Formally, we still need to check that the variance is not too high. Let X_k be the random variable that is 1 if a_k is one of the two largest elements among a_1,

$\ldots$, a_k, and 0 otherwise. Then $X = X_3 + X_4 + \ldots + X_N$. We have already proved that $\mathbb{E}[X_k] = 2/k$ and $\mathbb{E}[X] \approx 2\ln N$. Now we need to prove that with high probability X is not far away from $\mathbb{E}[X]$.

We will use Chebyshev's inequality:

$$\mathbb{P}[|X - \mathbb{E}[X]| \geq \Delta] \leq \frac{\text{Var}[X]}{\Delta^2}.$$

We have $\text{Var}[X] = \mathbb{E}[X^2] - \mathbb{E}[X]^2$. So we have to calculate $\mathbb{E}[X^2] = \sum_{i=3}^N \sum_{j=3}^N \mathbb{E}[X_i X_j]$.

Examine the random variable $X_i X_j$. If $i \neq j$, it is 1 only if $X_i = X_j = 1$, and 0 otherwise. Therefore, $\mathbb{E}[X_i X_j] = \mathbb{P}[X_i = 1, X_j = 1]$. If $i < j$, then for any choice of a_i such that $X_i = 1$, we still have at most 2 choices among $j-1$ choices for a_j such that $X_j = 1$. Therefore, $\mathbb{P}[X_i = 1, X_j = 1] \leq \frac{2}{i} \cdot \frac{2}{j-1} \leq \frac{2}{i-1} \cdot \frac{2}{j-1}$. When $i = j$, we have $\mathbb{E}[X_i X_i] = \mathbb{E}[X_i] = \frac{2}{i}$.

Hence,

$$\begin{aligned}
\mathbb{E}[X^2] &= \sum_{i=3}^{N} \frac{2}{i-1} \cdot \left(\left(\sum_{j=3}^{N} \frac{2}{j-1} \right) - \frac{2}{i-1} + \frac{i-1}{i} \right) \\
&\leq \sum_{i=3}^{N} \frac{2}{i-1} \cdot (2(\ln N + 1) + 1) = (2\ln N + 2)(2\ln N + 3)
\end{aligned}$$

by using $1/1 + 1/2 + \ldots + 1/N \leq \ln N + 1$. We also have

$$\mathbb{E}[X] = 2\left(\frac{1}{3} + \ldots + \frac{1}{N}\right) \geq 2\left(\ln N - \frac{3}{2}\right) = 2\ln N - 3.$$

by using $1/1 + 1/2 + \ldots + 1/N \geq \ln N$. Thus,

$$\text{Var}[X] \leq (2\ln N + 2)(2\ln N + 3) - (2\ln N - 3)^2 < 22\ln N.$$

Let $\Delta = 14\ln N$, then, by Chebyshev's inequality,

$$\mathbb{P}[|X - \mathbb{E}[X]| \geq \Delta] \leq \frac{22\ln N}{14^2 \ln^2 N} = \frac{22}{196\ln N},$$

which is less than 1% for $N = 99800$.

The total number of "additional case (b) queries" in the probable case is at most $2\ln N + \Delta = 16\ln N < 200$ for $N \leq 99800$.