



Incantation

Problem I

Luo Tsz Fung {pepper1208}

The 1st Buddhist Sin Tak College Computer Club Programming Contest

October 18, 2024



Background

Problem Idea by pepper1208

Preparation by pepper1208, rina__owo

stpc();



Problem Restatement

For the i -th spell cast, let the casting time be t_i seconds, and let the spell contain k_i individual spells. The IDs of these spells are denoted as $x_{i,1}, x_{i,2}, x_{i,3}, \dots, x_{i,k_i}$.

For each case, calculate the number of distinct spell IDs cast within the 24 hours preceding each spell casting time.

stpc();



Statistics

Points are given per checkpoint in this problem. You can get the point of the checkpoint when you pass them.

Attempts: 4

0 points 4 + 0 = 4

First solved by **No one!**

stpc();



Subtasks

Checkpoints	n	$\sum k_i \leq$	$x_{i,j} \leq$	$t_i \leq$
1 ~ 2	$= 1$	10	10	10
3 ~ 4	≤ 10	100	100	32767
5 ~ 8	≤ 100	100	100	86400
9 ~ 14	≤ 1000	3000	1000	10^9
15 ~ 20	$\leq 10^5$	3×10^5	10^5	10^9

stpc();



Checkpoints 1 ~ 2 (10 pts)

Sanity check.

Set up an array with the size of 10 (or more than 10 to reserve buffer) indicate the frequency of ID of different incantation.

For each cast, update the frequency of the array. After each update, perform linear search on the array to count the number of non-zero frequency.

Score: **10** (Cumulative Score: **10**)

stpc();



Checkpoints 3 ~ 8 (30 pts)

To reduce the time complexity, **discretization** could be performed.

As incantation ID could be very discrete, we can map those ID into a newly defined ID by us.

You can also use **hashmap** or **bucket** to implement the algorithm.

Score: **30** (Cumulative Score: **40**)

stpc();



Full Solution

From now on, there will be some casts that $t_i > 86400$. For each cast, we only care about casts that their time of casts falls in the range $[t_i - 86400, t_i]$.

Why don't we find the preceding cast in the valid range?

stpc();



Full Solution

Implement a time array that stores the time of different casts. Use a variable to store the index of the time array indicating the preceding cast in the valid range.

For every new cast, update the required variable if necessary.

Then, we can only perform processing of casts starting from the preceding cast that falls in the valid range!

Actually, this concept is related to a data structure **queue**.



Full Solution

By implementing the **queue concept** and the **frequency array**, the time complexity can be greatly reduced.

Score: **60** (Cumulative Score: **100**)

p.s. This is an OI-like problem which may appear in HKOI or CSP. Pay attention, OI team members!

stpc();



Takeaways

1. Be brave to try different possible solution in your mind.
2. A solid foundation of your coding skills is required for harder question. Your idea should not be restricted by coding skills.
3. Learn to give up.

stpc();

