

Analysis: Crime House

You are given a log of criminals entering and leaving a house from the front door. The log is for one day, and at the start of the day there might be criminals in the house. This house may have other entrances and exits besides the front door, and the criminals might wear masks when entering/leaving the house. The task is to figure out if, given the log, it is possible that there is only one door (i.e. the front door only), and if so to figure out the minimum number of people that could be in the house at the end of the day.

Building intuition:

Before proceeding with the solution, let us develop some intuition for the problem. Let us say we are actually given a log where **none** of the criminals wore a mask, i.e. none of the entries are $\sim E 0$ or $\sim L 0$. In that case, the solution would be to simulate the log from **start to end**, i.e. simulate criminals entering and leaving the house via **one** door only, and to see if the simulation is valid. When is a simulation valid? Let us instead discuss when a simulation would be invalid. During the simulation, if we encounter a $\sim E X$ command (note $X > 0$), and $\sim X$ was already in the house then it is invalid (impossible to enter twice without leaving). Similarly if we encounter a $\sim L X$ command, and $\sim X$ isn't already in the house then it is invalid. Note that if it was the first time encountering the $\sim L X$ command then we can say that $\sim X$ was already in the house at the start of the day so it would be valid.

The reason why the problem we are given is complicated is because of the presence of masked criminals. We are going to tackle this problem using a greedy strategy. We will first assume that there are $\sim S$ number of criminals at the start of the day. Then we will do a simulation like we described before (i.e. simulation from start to end), except whenever we encounter a masked criminal, we will try to greedily assign a criminal number to the masked criminal. We will describe this greedy assignment later, but first let us describe a few terms we use in the editorial.

Terms:

Let us first describe some of the terms we use in this editorial. We call the log an **event** queue, which is a sequence of $\sim E 0$, $\sim L 0$, $\sim E X$, or $\sim L X$ (where $X > 0$) events. During our simulation, we will go through the queue from start to end. During the simulation let us say we are at index i in the event queue, then we define the **current** event queue to be the subsequence of the event queue starting from index i to the end of the queue. During the simulation, we will also maintain a set of criminal numbers that are inside the house: we call the set **INSIDE**. Finally, the last term to define is the **next known event** for a particular criminal (non-zero) number. To help understand this term, we will use an example. Let us say our current event queue (along with the location indices) is:

```
0:  $\sim E 0$ 
1:  $\sim E 0$ 
2:  $\sim E 5$ 
3:  $\sim L 1$ 
4:  $\sim E 1$ 
5:  $\sim L 1$ 
6:  $\sim E 1$ 
7:  $\sim L 5$ 
8:  $\sim E 2$ 
```

There are three unmasked criminal numbers: 1, 2 and 5. Note we do not consider the masked criminals. Criminal \hat{c}_1 appears 4 times in the current queue at indices 3, 4, 5, and 6. The next known event for criminal \hat{c}_1 is defined to be \hat{L}_1 at index 3 (which is the earliest among the 4 events). Similarly criminal \hat{c}_2 appears 1 time at index 8, therefore the next known event for criminal \hat{c}_2 is \hat{E}_2 at index 8. Similarly criminal \hat{c}_5 appears 2 times, at indices 2 and 7. The next known event for criminal \hat{c}_5 is defined to be \hat{E}_5 at index 2.

Greedy strategy:

Now, let us proceed with the explanation for the greedy solution. First, let us assume we start with \hat{S} criminals already in the house. We will simulate the presence of these \hat{S} criminals in the house by adding \hat{S} \hat{E}_0 events at the start of the event queue. To make things simpler during the simulation, we will pretend that all the criminals leave at the end of the day. To do so, we can add \hat{T} \hat{L}_0 events to the end of the event queue such that the number of \hat{E} events and \hat{L} events are equal. We will then proceed to simulate this resulting event queue to check for validity and also in the process, if possible, assign numbers to the masked criminals. If the simulation is a valid simulation then for this event queue, our answer is \hat{T} i.e. the number of criminals remaining in the house at the end of the day. We will discuss how we find \hat{S} (and correspondingly \hat{T}) later.

We now describe the simulation. As mentioned earlier, we simulate from the start to the end. If we are able to successfully go through all the events then this event queue is a valid event queue. At any instant we have a current event queue (which is defined above). Note that at the start of the simulation, the set INSIDE is an empty set. We will now discuss how to address the four cases for the first event in the current queue, i.e. the cases \hat{E}_X , \hat{L}_X , \hat{E}_0 and \hat{L}_0 (note that $X > 0$).

Case \hat{E}_X :

In this case, person \hat{X} is entering the house. It will only be valid if \hat{X} is *not* already inside the house, which we can check by seeing if the INSIDE set does not already contain \hat{X} . If true, then this is a valid event, else it is an invalid event hence an invalid simulation therefore we end the simulation. Also, if \hat{E}_X is valid, we need to update the INSIDE set by adding \hat{X} to it.

Case \hat{L}_X :

In this case, person \hat{X} is leaving the house. It will only be valid if \hat{X} is already inside the house, which we can check by seeing if the INSIDE set contains \hat{X} . If true, then this is a valid event, else it is an invalid event hence an invalid simulation therefore we end the simulation. Also, if \hat{L}_X is valid, we need to update the INSIDE set by removing \hat{X} from it.

Case \hat{E}_0 :

In this case, a masked criminal is entering. Our objective is to assign a number to this masked criminal. If we are able to assign a number to the masked criminal, the criminal will then enter the house which we do so by adding the assigned criminal number to the INSIDE set. Now the question is how do we assign numbers to this masked criminal. There are two cases to consider:

Case \hat{E}_0 (a):

First, we consider the set of known criminals (i.e. ones that aren't masked) in the current queue who are also not inside the house (i.e. not in the INSIDE set). Out of these criminals, we consider the criminals for which the next known event is a leave event (i.e. \hat{X} for criminal \hat{X}). These are the people who need to leave the house at some point, but it isn't known when they leave the house. All of these people must enter the house in \hat{E} events that happen before they leave. If such a person exists, we make a greedy choice and choose the person who is leaving soonest, to leave ourselves with maximum time to get the remaining people out. Why do we make this greedy choice? Consider two such people, 1 and 2. The sequence, then, looks like this:

\hat{E} $\hat{0}$... \hat{L} $\hat{1}$... \hat{L} $\hat{2}$

Note that the \hat{E} ... \hat{L} denotes other events in the event queue which we do not list out to avoid complexity. In order for this event queue to be valid, there must be another \hat{E} $\hat{0}$ to match the \hat{L} $\hat{1}$ and \hat{L} $\hat{2}$ events (note that 1 and 2 are not in the INSIDE set). There are two possibilities then:

\hat{E} $\hat{0}$... \hat{E} $\hat{0}$... \hat{L} $\hat{1}$... \hat{L} $\hat{2}$

and

\hat{E} $\hat{0}$... \hat{L} $\hat{1}$... \hat{E} $\hat{0}$... \hat{L} $\hat{2}$

In both situations, choosing the entering masked criminal to have been criminal number $\hat{1}$ works. But if we had instead chosen the entering masked criminal to have been criminal number $\hat{2}$, then the second possibility listed above would be invalid (the second \hat{L} $\hat{1}$ cannot be matched with the \hat{E} $\hat{0}$ that comes after it). Therefore we are better off if we choose $\hat{1}$. This logic generalizes to the case where there are more than two people.

Case \hat{E} $\hat{0}$ (b):

If no such person in Case \hat{E} $\hat{0}$ (a) exists, we simply create a new person who doesn't show up anywhere unmasked (i.e. assign a number that is big e.g. a million; make sure to assign unique numbers to each new person). Why create a new person? It is because the alternative is to use someone who is outside the building, but whose next known event is an enter event. For example, let's suppose the person doing that is person 5. Then we have a sequence like this:

\hat{E} $\hat{0}$... \hat{E} $\hat{5}$

For person 5 to be entering at the start, we need the sequence to look like this:

\hat{E} $\hat{0}$... \hat{L} $\hat{0}$... \hat{E} $\hat{5}$

In that situation, having person 1,000,000 enter is just as good as having person 5 enter. But in the situation where there is no \hat{L} $\hat{0}$ between the \hat{E} $\hat{0}$ and the \hat{E} $\hat{5}$, person 1,000,000 could leave the house later (if there is a \hat{L} $\hat{0}$ afterwards), but person 5 is stuck entering the house, then entering it again causing the simulation to be invalid. Therefore we better off to create a new person with a new unique number in such cases.

Case \hat{L} $\hat{0}$:

In this case, a masked criminal is leaving the house. As in the case for case \hat{E} $\hat{0}$, the goal is to assign a number to the masked criminal. After assigning the number \hat{X} to the criminal (if that is possible), we must update the INSIDE set by removing \hat{X} from the set. Note that if the INSIDE set is empty when we encounter the \hat{L} $\hat{0}$ event then it is an

invalid simulation. For cases when there are criminals in the INSIDE set, there are three cases to consider:

Case $\hat{L} \hat{0}$ (a):

Consider all the criminals who are inside, and whose next known event is that they are entering the house. By the same logic as in Case $\hat{E} \hat{0}$ (a), we choose the one who is entering the soonest. If we are able to do such an assignment, we need not consider any of the other \hat{L} cases.

Case $\hat{L} \hat{0}$ (b):

If no such person in the previous case exists, then we consider all criminals who are inside the house. If there is one criminal who has no next known event (either $\hat{E} \hat{X}$ or $\hat{L} \hat{X}$) with the house, have that person leave; they have to leave at some point in no particular order, and it's better to have them leave than someone whose next known event is that they are leaving. If we were able to choose such a criminal, then we need not consider the last case.

Case $\hat{L} \hat{0}$ (c):

If no such person in the previous two cases exists, we're now stuck with criminals who are inside and whose next known event is that they're leaving. In such a case, we make a greedy choice and take the one who leaves as late as possible, so we have the maximum number of chances for our selected person to enter the house again (with an $\hat{E} \hat{0}$ event). Why do we make this greedy choice? Consider two such people we need to choose from, 1 and 2. The sequence, then, looks like this:

$\hat{L} \hat{0} \dots \hat{L} \hat{1} \dots \hat{L} \hat{2}$

In order for this to be valid, there must be another $\hat{E} \hat{0}$. There are two possibilities:

$\hat{L} \hat{0} \dots \hat{E} \hat{0} \dots \hat{L} \hat{1} \dots \hat{L} \hat{2}$

and

$\hat{L} \hat{0} \dots \hat{L} \hat{1} \dots \hat{E} \hat{0} \dots \hat{L} \hat{2}$

In both situations, assigning the leaving masked person ($\hat{L} \hat{0}$) to be person 2 works, but choosing 1 does not work for the second possibility. So we are better off to choose person 2 for $\hat{L} \hat{0}$. This logic generalizes to the case where there are more than two people.

And so that concludes the four cases to consider for the first event in the current queue. As discussed earlier, if the first event is deemed invalid when considering the four cases, then we have an invalid simulation. But if we are able to successfully assign a criminal number, then we advance to the next event in the event queue and get a new current queue, and continue on with our simulation. If we reach the end, then it is a valid event queue.

Picking \hat{S} criminals initially in the house:

Earlier on, we had assumed there were \hat{S} criminals initially in the house. But the question is what value do we consider for \hat{S} ? We can try all values from 0 to N (the size of the queue). Doing so one by one might still be slow for the large input. However, we can use binary search to make the search faster. Note that having more people (larger \hat{S}) in the house gives us more flexibility to make the simulation work i.e. bigger values of \hat{S} gives

us more \hat{E}^0 at the start and more \hat{L}^0 at the end to work with. Also if the simulation was valid for $\hat{S}^T = i$, it would be valid for $\hat{S}^T = i + 1$ too because we are adding a \hat{E}^0 at the start and correspondingly a \hat{L}^0 at the end (which can match each other in our already valid simulation for $\hat{S}^T = i$). This means that for the smaller values of \hat{S}^T , we might get invalid simulations (i.e. false values), but after a certain \hat{S}^T value we will start getting valid simulations (i.e. true values). This indicates that the boolean function (i.e. that function that indicates whether the simulation is valid or not) is a [monotonic boolean function](#) (for input $\hat{S}^T = 0$ to N). When a function is monotonic, we can use [binary search](#) to make the search faster.

Finally, as specified in the problem, if we are unable to find any valid simulation for any of the values of \hat{S}^T then we report CRIME TIME as the answer.