Problem 3433: Count Mentions Per User

Problem Information

Difficulty: Medium
Acceptance Rate: 30.52%
Paid Only: No
Tags: Array, Math, Sorting, Simulation

Problem Description

You are given an integer `numberOfUsers` representing the total number of users and an array `events` of size `n x 3`.

Each `events[i]` can be either of the following two types:

1. **Message Event:** `["MESSAGE", "timestampi", "mentions_stringi"]` * This event indicates that a set of users was mentioned in a message at `timestampi`. * The `mentions_stringi` string can contain one of the following tokens: * `id<number>`: where `<number>` is an integer in range `[0,numberOfUsers - 1]`. There can be **multiple** ids separated by a single whitespace and may contain duplicates. This can mention even the offline users. * `ALL`: mentions **all** users. * `HERE`: mentions all **online** users. 2. **Offline Event:** `["OFFLINE", "timestampi", "idi"]` * This event indicates that the user `idi` had become offline at `timestampi` for **60 time units**. The user will automatically be online again at time `timestampi + 60`.

Return an array `mentions` where `mentions[i]` represents the number of mentions the user with id `i` has across all `MESSAGE` events.

All users are initially online, and if a user goes offline or comes back online, their status change is processed _before_ handling any message event that occurs at the same timestamp.

Note that a user can be mentioned **multiple** times in a **single** message event, and each mention should be counted **separately**.

Example 1:

Input: numberOfUsers = 2, events = [["MESSAGE","10","id1 id0"],["OFFLINE","11","0"],["MESSAGE","71","HERE"]]

Output: [2,2]

Explanation:

Initially, all users are online.

At timestamp 10, `id1` and `id0` are mentioned. `mentions = [1,1]`

At timestamp 11, `id0` goes **offline.**

At timestamp 71, `id0` comes back **online** and `"HERE"` is mentioned. `mentions = [2,2]`

Example 2:

Input: numberOfUsers = 2, events = [["MESSAGE","10","id1 id0"],["OFFLINE","11","0"],["MESSAGE","12","ALL"]]

Output: [2,2]

Explanation:

Initially, all users are online.

At timestamp 10, `id1` and `id0` are mentioned. `mentions = [1,1]`

At timestamp 11, `id0` goes **offline.**

At timestamp 12, `"ALL"` is mentioned. This includes offline users, so both `id0` and `id1` are mentioned. `mentions = [2,2]`

Example 3:

Input: numberOfUsers = 2, events = [["OFFLINE","10","0"],["MESSAGE","12","HERE"]]

Output: [0,1]

Explanation:

Initially, all users are online.

At timestamp 10, `id0` goes **offline.**

At timestamp 12, `"HERE"` is mentioned. Because `id0` is still offline, they will not be mentioned. `mentions = [0,1]`

Constraints:

* `1 <= numberOfUsers <= 100` * `1 <= events.length <= 100` * `events[i].length == 3` * `events[i][0]` will be one of `MESSAGE` or `OFFLINE`. * `1 <= int(events[i][1]) <= 105` * The number of `id<number>` mentions in any `"MESSAGE"` event is between `1` and `100`. * `0 <= <number> <= numberOfUsers - 1` * It is **guaranteed** that the user id referenced in the `OFFLINE` event is **online** at the time the event occurs.

Code Snippets

C++:

```
class Solution {
public:
vector<int> countMentions(int numberOfUsers, vector<vector<string>>& events)
{

}
};
```

Java:

```
class Solution {
public int[] countMentions(int numberOfUsers, List<List<String>> events) {

}
}
```

Python3:

```
class Solution:
def countMentions(self, numberOfUsers: int, events: List[List[str]]) ->
List[int]:
```