LeetCode
Explore
Problems
New
Interview
Contest
Discuss
Store
🎁 LeetCoding Challenge + GIVEAWAY! 🎁
Description
Solution
Discuss (570)
Submissions
Java
Autocomplete
Hard
1167
74
Add to List
Share
Yes
No
Problems
Pick One
Prev
/99
Next
Console
Contribute
Run Code
Submit

759. Employee Free Time

We are given a list `schedule` of employees, which represents the working time for each employee.

Each employee has a list of non-overlapping `Intervals`, and these intervals are in sorted order.

Return the list of finite intervals representing **common, positive-length free time** for *all* employees, also in sorted order.

(Even though we are representing `Intervals` in the form `[x, y]`, the objects inside are `Intervals`, not lists or arrays. For example, `schedule[0][0].start = 1`, `schedule[0][0].end = 2`, and `schedule[0][0][0]` is not defined). Also, we wouldn't include intervals like [5, 5] in our answer, as they have zero length.

Example 1:

```
Input: schedule = [[[1,2],[5,6]],[[1,3]],[[4,10]]]
Output: [[3,4]]
Explanation: There are a total of three employees, and all common
free time intervals would be [-inf, 1], [3, 4], [10, inf].
We discard any intervals that contain inf as they aren't finite.
```

Example 2:

```
Input: schedule = [[[1,3],[6,7]],[[2,4]],[[2,5],[9,12]]]
Output: [[5,6],[7,9]]
```

Constraints:

- `1 <= schedule.length , schedule[i].length <= 50`
- `0 <= schedule[i].start < schedule[i].end <= 10^8`

Accepted 89,095 | Submissions 126,804

Seen this question in a real interview before?

Companies

0 ~ 6 months | 6 months ~ 1 year | 1 year ~ 2 years

Amazon | 8 Pinterest | 7 Wayfair | 6 Karat | 6 DoorDash | 5 Apple | 5 Google | 4 Airbnb | 4 Uber | 3 Facebook | 3 Oracle | 2

Related Topics

Array Sorting Heap (Priority Queue)

Similar Questions

Merge Intervals	Medium
Interval List Intersections	Medium

Hide Hint 1

Take all the intervals and do an "events" (or "line sweep") approach - an event of (x, OPEN) increases the number of active intervals, while (x, CLOSE) decreases it. Processing in sorted order from left to right, if the number of active intervals is zero, then you crossed a region of common free time.

```java
/*
// Definition for an Interval.
class Interval {
    public int start;
    public int end;

    public Interval() {}

    public Interval(int _start, int _end) {
        start = _start;
        end = _end;
    }
};
*/

class Solution {
    public List<Interval> employeeFreeTime(List<List<Interval>> schedule) {
        
    }
}
```