Problem 759: Employee Free Time

Problem Information

Difficulty: Hard
Acceptance Rate: 72.64%
Paid Only: Yes
Tags: Array, Line Sweep, Sorting, Heap (Priority Queue)

Problem Description

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`

Code Snippets

C++:

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

Interval() {}

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

class Solution {
public:
vector<Interval> employeeFreeTime(vector<vector<Interval>> schedule) {

}
};
```

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) {

}
}
```

Python3:

```
"""
# Definition for an Interval.
class Interval:
def __init__(self, start: int = None, end: int = None):
self.start = start
self.end = end
"""

class Solution:
def employeeFreeTime(self, schedule: '[[Interval]]') -> '[Interval]':
```