Problem 2589: Minimum Time to Complete All Tasks

Problem Information

Difficulty: Hard
Acceptance Rate: 38.98%
Paid Only: No
Tags: Array, Binary Search, Stack, Greedy, Sorting

Problem Description

There is a computer that can run an unlimited number of tasks **at the same time**. You are given a 2D integer array `tasks` where `tasks[i] = [starti, endi, durationi]` indicates that the `ith` task should run for a total of `durationi` seconds (not necessarily continuous) within the **inclusive** time range `[starti, endi]`.

You may turn on the computer only when it needs to run a task. You can also turn it off if it is idle.

Return _the minimum time during which the computer should be turned on to complete all tasks_.

Example 1:

Input: tasks = [[2,3,1],[4,5,1],[1,5,2]] **Output:** 2 **Explanation:** - The first task can be run in the inclusive time range [2, 2]. - The second task can be run in the inclusive time range [5, 5]. - The third task can be run in the two inclusive time ranges [2, 2] and [5, 5]. The computer will be on for a total of 2 seconds.

Example 2:

Input: tasks = [[1,3,2],[2,5,3],[5,6,2]] **Output:** 4 **Explanation:** - The first task can be run in the inclusive time range [2, 3]. - The second task can be run in the inclusive time ranges [2, 3] and [5, 5]. - The third task can be run in the two inclusive time range [5, 6]. The computer will be on for a total of 4 seconds.

Constraints:

* `1 <= tasks.length <= 2000` * `tasks[i].length == 3` * `1 <= starti, endi <= 2000` * `1 <= durationi <= endi - starti + 1 `

Code Snippets

C++:

```
class Solution {
public:
int findMinimumTime(vector<vector<int>>& tasks) {

}
};
```

Java:

```
class Solution {
public int findMinimumTime(int[][] tasks) {

}
}
```

Python3:

```
class Solution:
def findMinimumTime(self, tasks: List[List[int]]) -> int:
```