Store (/subscribe? ref=nb_npl)

Back to Contest (/contest/weekly-contest-336/)



6318. Minimum Time to Complete All Tasks

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 $[i] = [start_i, end_i, duration_i]$ indicates that the ith task should run for a total of $duration_i$ seconds (not necessarily continuous) within the **inclusive** time range $[start_i, end_i]$.

My Submissions (/contest/weekly-contest-336/problems/minimum-time-to-complete-all-tasks/submissions/)

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.

User Accepted: 0 User Tried: 0 Total Accepted: 0 Total Submissions: 0 Difficulty: Hard

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</li>
tasks[i].length == 3
1 <= start<sub>i</sub>, end<sub>i</sub> <= 2000</li>
1 <= duration<sub>i</sub> <= end<sub>i</sub> - start<sub>i</sub> + 1
```

```
JavaScript •
```

```
1 v const findMinimumTime = (a) ⇒ {
2
        let on = Array(2005).fill(false), res = 0;
 3
        a.sort((x, y) \Rightarrow x[1] - y[1]);
 4
        for (const [l, r, t] of a) {
             let cur = 0;
 5
 6
             for (let i = l; i <= r; i++) cur += on[i];
 7
             let curT = Math.max(0, t - cur);
 8 •
             for (let i = r; i >= l; i--) {
9 •
                 if (!on[i] && curT > 0) {
10
                     on[i] = true;
11
                     curT--;
12
                     res++;
13
                 }
14
             }
15
16
        return res;
17
    };
```

₁ C

Custom Testcase Use Example Testcases

Submission Result: Accepted (/submissions/detail/913607728/)

More Details ➤ (/submissions/detail/913607728/)

Share your acceptance!

Copyright © 2023 LeetCode

Help Center (/support) | Jobs (/jobs) | Bug Bounty (/bugbounty) | Online Interview (/interview/) | Students (/student) | Terms (/terms) | Privacy Policy (/privacy)

United States (/region)