

[Back to Contest \(/contest/weekly-contest-336/\)](/contest/weekly-contest-336/)

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 |

- 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.

- 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.

- $1 \leq \text{tasks.length} \leq 2000$
- $\text{tasks}[i].\text{length} = 3$
- $1 \leq \text{start}_i, \text{end}_i \leq 2000$
- $1 \leq \text{duration}_i \leq \text{end}_i - \text{start}_i + 1$





```

1 ▽ const findMinimumTime = (a) => {
2     let on = Array(2005).fill(false), res = 0;
3     a.sort((x, y) => x[1] - y[1]);
4     for (const [l, r, t] of a) {
5         let cur = 0;
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 };

```

☐ Custom Testcase

Use Example Testcases

Run

Submit

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

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

Share your acceptance!