6318. Minimum Time to Complete All Tasks

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

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 `i^th` task should run for a total of `duration_i` seconds (not necessarily continuous) within the **inclusive** time range `[start_i, end_i]`.

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:	462
User Tried:	1738
Total Accepted:	537
Total Submissions:	4162
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`
- `tasks[i].length == 3`
- `1 <= start_i, end_i <= 2000`
- `1 <= duration_i <= end_i - start_i + 1`

Python

```python
class Solution(object):
    def findMinimumTime(self, tasks):
        """
        :type tasks: List[List[int]]
        :rtype: int
        """
        start = 100000
        end = -1
        for i in range(len(tasks)):
            if tasks[i][0] < start:
                start = tasks[i][0]
            if tasks[i][1] > end:
                end = tasks[i][1]

        countInstance = []
        countInstance2 = []
        for i in range(start, end+1, 1):
            countInstance.append(0)
            countInstance2.append(0)

        #print(countInstance)

        for i in range(len(tasks)):
            for j in range(tasks[i][0], tasks[i][1]+1, 1):
                countInstance[j-1] += 1

```

```
        for i in range(len(tasks)):
            for j in range(tasks[i][0], tasks[i][1]+1, 1):


        val = max(countInstance)
        print(val)
        print(countInstance.count(val))
        print(countInstance.count(val)/2)
        return countInstance.count(val)/2 # I give up
```

Custom Testcase | Use Example Testcases

Run | Submit