

Problem Challenge 2

We'll cover the following ^

- Scheduling Tasks (hard)
- Try it yourself

Scheduling Tasks (hard)

You are given a list of tasks that need to be run, in any order, on a server. Each task will take one CPU interval to execute but once a task has finished, it has a cooling period during which it can't be run again. If the cooling period for all tasks is 'K' intervals, find the minimum number of CPU intervals that the server needs to finish all tasks.

If at any time the server can't execute any task then it must stay idle.

Example 1:


```
Input: [a, a, a, b, c, c], K=2
Output: 7
Explanation: a -> c -> b -> a -> c -> idle -> a
```


Example 2:


```
Input: [a, b, a], K=3
Output: 5
Explanation: a -> b -> idle -> idle -> a
```


Try it yourself

Try solving this question here:

 Java

 Python3

 JS


 C++

```
1 def schedule_tasks(tasks, k):
2     intervalCount = 0
3     # TODO: Write your code here
4     return intervalCount
5
6
7 def main():
8     print("Minimum intervals needed to execute all tasks: " +
9         str(schedule_tasks(['a', 'a', 'a', 'b', 'c', 'c'], 2)))
10    print("Minimum intervals needed to execute all tasks: " +
11        str(schedule_tasks(['a', 'b', 'a'], 3)))
12
13
14 main()
15
16
```

Run

Save

Reset



[← Back](#)[Next →](#)

