Problem Challenge 2

We'll cover the following

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++

```java
import java.util.*;

class TaskScheduler {

  public static int scheduleTasks(char[] tasks, int k) {
    // TODO: Write your code here
    return -1;
  }

  public static void main(String[] args) {
    char[] tasks = new char[] { 'a', 'a', 'a', 'b', 'c', 'c' };
    System.out.println("Minimum intervals needed to execute all tasks: " + TaskScheduler.scheduleTasks(ta

    tasks = new char[] { 'a', 'b', 'a' };
    System.out.println("Minimum intervals needed to execute all tasks: " + TaskScheduler.scheduleTasks(ta
  }
}
```

Run | Save | Reset

Back

Solution Review: Problem Challenge 1

Next

Solution Review: Problem Challenge 2

Mark as Completed