[LeetCode](https://leetcode.com/problems/task-scheduler/submissions/1207856556/?envType=daily-question&envId=2024-03-19)

<https://github.com/AdityaKonda6/-50DaysOfCoding>

<https://leetcode.com/problems/task-scheduler/submissions/1207856556/?envType=daily-question&envId=2024-03-19>

<https://www.linkedin.com/in/aditya-adi-konda/>

Day 29 of [#50dayscodingchallenge](https://www.linkedin.com/feed/hashtag/?keywords=50dayscodingchallenge&highlightedUpdateUrns=urn:li:activity:7166316239483461633):  
[#leetcode](https://www.linkedin.com/feed/hashtag/?keywords=leetcode&highlightedUpdateUrns=urn:li:activity:7166316239483461633) [#leetcodechallenge](https://www.linkedin.com/feed/hashtag/?keywords=leetcodechallenge&highlightedUpdateUrns=urn:li:activity:7166316239483461633) [#leetcodestreak](https://www.linkedin.com/feed/hashtag/?keywords=leetcodestreak&highlightedUpdateUrns=urn:li:activity:7166316239483461633) [#leetcode2024](https://www.linkedin.com/feed/hashtag/?keywords=leetcode2024&highlightedUpdateUrns=urn:li:activity:7166316239483461633) [#leetcode50day](https://www.linkedin.com/feed/hashtag/?keywords=leetcode50day&highlightedUpdateUrns=urn:li:activity:7166316239483461633)  
   
Ventured further into my coding journey today, tackling the engaging LeetCode Problem "Successfully solved LeetCode Problem 🎈💻“621. Task Scheduler.”  
   
✨ Task: You are given an array of CPU tasks, each represented by letters A to Z, and a cooling time, n. Each cycle or interval allows the completion of one task. Tasks can be completed in any order, but there's a constraint: identical tasks must be separated by at least n intervals due to cooling time.

Return the minimum number of intervals required to complete all tasks.

Examples:

Example 1:

Input: tasks = ["A","A","A","B","B","B"], n = 2

Output: 8

Explanation: A possible sequence is: A -> B -> idle -> A -> B -> idle -> A -> B.

After completing task A, you must wait two cycles before doing A again. The same applies to task B. In the 3rd interval, neither A nor B can be done, so you idle. By the 4th cycle, you can do A again as 2 intervals have passed.

Example 2:

Input: tasks = ["A","C","A","B","D","B"], n = 1

Output: 6

Explanation: A possible sequence is: A -> B -> C -> D -> A -> B.

With a cooling interval of 1, you can repeat a task after just one other task.

Example 3:

Input: tasks = ["A","A","A", "B","B","B"], n = 3

Output: 10

Explanation: A possible sequence is: A -> B -> idle -> idle -> A -> B -> idle -> idle -> A -> B.

There are only two types of tasks, A and B, which need to be separated by 3 intervals. This leads to idling twice between repetitions of these tasks.

Let's Connect:

If you find this problem intriguing or have insights to share, let's connect! I'm passionate about problem-solving, algorithmic thinking, and collaborative learning. Feel free to comment or reach out for engaging discussions and knowledge exchange.Unravel the mystery using your coding skills!

[#CodingChallenge](https://www.linkedin.com/feed/hashtag/?keywords=codingchallenge&highlightedUpdateUrns=urn:li:activity:7166316239483461633) [#Algorithm](https://www.linkedin.com/feed/hashtag/?keywords=algorithm&highlightedUpdateUrns=urn:li:activity:7166316239483461633) [#LinkedInPost](https://www.linkedin.com/feed/hashtag/?keywords=linkedinpost&highlightedUpdateUrns=urn:li:activity:7166316239483461633) #Algorithm #Optimization #DataStructures #CodingChallenge  
  
Excited about the progress and challenges ahead!  
   
Make Sure You Follow My GitHub For Solutions: <https://github.com/AdityaKonda6/-50DaysOfCoding>  
  
  
Happy coding!

**Solution:-**

class Solution {

  public int leastInterval(char[] tasks, int n) {

    int[] count = new int[26];

    for (final char task : tasks)

      ++count[task - 'A'];

    final int maxFreq = Arrays.stream(count).max().getAsInt();

    final int maxFreqTaskOccupy = (maxFreq - 1) \* (n + 1);

    final int nMaxFreq = (int) Arrays.stream(count).filter(c -> c == maxFreq).count();

    return Math.max(maxFreqTaskOccupy + nMaxFreq, tasks.length);

  }

}

