Given a char array representing tasks CPU need to do. It contains capital letters A to Z where different letters represent different tasks. Tasks could be done without original order. Each task could be done in one interval. For each interval, CPU could finish one task or just be idle.

However, there is a non-negative cooling interval **n** that means between two **same tasks**, there must be at least n intervals that CPU are doing different tasks or just be idle.

You need to return the **least** number of intervals the CPU will take to finish all the given tasks.

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

**Output:** 8

**Explanation:** A -> B -> idle -> A -> B -> idle -> A -> B.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 思想：  计算出现次数最多的字母的出现次数，然后可以拼成下面这样的，宽度为n+1，为什么高度为（次数-1）呢，因为最后一行，只要把字母都填进去了，统计时长过程就结束了，后面有多少空格，都不会纳入总时间。  遍历每个字母出现的次数，空格数- Math.min(高度，字母出现的次数)。  这里为啥要用min呢，因为比如B也出现了4次的话，此时高度是3，这个时候B其实最多也就占用了我们空格里的{高度}个，最后剩下的那个在第{i+1}行，根本没算在我们画出来的这个空格表里。   |  |  |  | | --- | --- | --- | | A |  |  | | A |  |  | | A |  |  | | A |   如果现在所有空格都减光了，那么消耗的时间就是：全部字母个数，没有时间间隔  如果现在空格没有被填满，剩下的空格+tasks的长度即消耗的时间。 |
| 思考点：  这个题，有一个数学的点在于，只要知道出现次数最多的字母，出现几次就够了。  因为有多少种字母其实没关系，字母数越多，越可以灵活的塞进一次cycle里。  如果字母种类>n，那证明2次满满的cycle里可能有不同的字母填充，间接保证出现次数多的字母可以隔开。  如果字母种类<n，那上面的空格就会有空 |