

621 任务调度器

621. 任务调度器

难度 中等 362 收藏 分享 评论 举报

给定一个用字符数组表示的 CPU 需要执行的任务列表。其中包含使用大写的 A - Z 字母表示的 26 种不同种类的任务。任务可以以任意顺序执行，并且每个任务都可以在 1 个单位时间内执行完。CPU 在任何一个单位时间内都可以执行一个任务，或者在待命状态。

然而，两个相同种类的任务之间必须有长度为 n 的冷却时间，因此至少有连续 n 个单位时间内 CPU 在执行不同的任务，或者在待命状态。

你需要计算完成所有任务所需要的最短时间。

示例：

输入：tasks = ["A","A","A","B","B","B"], n = 2
输出：8
解释：A -> B -> (待命) -> A -> B -> (待命) -> A -> B。
在本示例中，两个相同类型任务之间必须间隔长度为 $n = 2$ 的冷却时间，而执行一个任务只需要一个单位时间，所以中间出现了 (待命) 状态。

解题思路：

- 1 统计每个任务的次数
- 2 找到最多的那个，根据其数量判断，最少需要时间

```
1 res = (max_task_cout - 1) * (n + 1)
```

3 判断，现有的任务数是否比这个少

如果少，则res就是最小值

如果多，那么任务数本身

代码：

```
1 def leastInterval(self, tasks: List[str], n: int) -> int:
2     length = len(tasks)
3     %23 记录每个任务出现的次数
4     task_map = collections.Counter(tasks)
5     %23 按任务出现的次数从大到小排序
6     task_sort = sorted(task_map.items(), key=lambda x:x[1], reverse=True)
7     %23 出现最多次任务的次数
8     max_task_cout = task_sort[0][1]
9     %23 计算最少需要的时间:
10    %23 A 空格 空格 A 空格 空格
11    res = (max_task_cout - 1) * (n + 1)
```

```
13         %23 计算把任务都加入，则剩余多少
14         for sort in task_sort:
15             res -= min(sort[1], max_task_cout-1)
16         %23 求其最大值，如果是负数则就是0， 如果是正数，就是其本身
17         res = max(0, res)
18         return res + length
```

时间复杂度 $O(n \log n)$

空间复杂度 $O(n)$