

切换主题: 默认主题

题目地址(1904. 你完成的完整对局数)



https://leetcode-cn.com/problems/the-number-of-full-rounds-you-have-played/

题目描述

一款新的在线电子游戏在近期发布，在该电子游戏中，以 刻钟 为周期规划若干时长为 15 分钟 的游戏对局。这意味着，在 HH:00、HH:15、HH:30 和 HH:45 四个时刻，新的对局生成。你进入游戏对局的时间为 startTime ，游戏结束的时间为 finishTime 。给你两个字符串 startTime 和 finishTime ，均符合 "HH:MM" 格式，分别表示你 进入 和 退出 游戏的确切时间，请计算在整个游戏会话期间，你完成的完整对局的对局数。

例如，如果 startTime = "05:20" 且 finishTime = "05:59" ，这意味着你仅仅完成从 05:30 到 05:45 这一个完整对局。而你并没有完成从 05:15 到 05:30 的对局，因为你没有在 05:15 进入游戏。

如果 finishTime 早于 startTime ，这表示你玩了个通宵（也就是从 startTime 到午夜，再从午夜到 finishTime ）。

假设你是从 startTime 进入游戏，并在 finishTime 退出游戏，请计算并返回你完成的 完整对局的对局数 。

示例 1:

输入: startTime = "12:01", finishTime = "12:44"

输出: 1

解释: 你完成了从 12:15 到 12:30 的一个完整对局。

你没有完成从 12:00 到 12:15 的完整对局，因为你是从对局开始后的 12:01 进入的游戏。

你没有完成从 12:30 到 12:45 的完整对局，因为你是从对局结束前的 12:44 退出的游戏。

示例 2:

输入: startTime = "20:00", finishTime = "06:00"

输出: 40

解释: 你完成了从 20:00 到 00:00 的 16 个完整的对局，以及从 00:00 到 06:00 的 24 个完整的对局。

16 + 24 = 40

示例 3:

输入: startTime = "00:00", finishTime = "23:59"

输出: 95

解释: 除最后一个小时你只完成了 3 个完整对局外，其余每个小时均完成了 4 场完整对局。

提示:

startTime 和 finishTime 的格式为 HH:MM

00 <= HH <= 23

00 <= MM <= 59

startTime 和 finishTime 不相等

前置知识

- 模拟

标签

- 模拟

难度

- 中等

入选理由

- 模拟题有啥难的？难就难在代码怎么写。这道题就用了一个技巧使得代码写起来简单许多。

公司

- 暂无

思路

我们可以将开始时间和结束时间先进行一次规范化处理，这样可以减少判断。

具体来说，我们可以对开始时间的分数进行如下处理：

- 如果开始时间的分数在 (0,15) 之间，那么可以等价于在 15 分开始，因此可以将开始时间直接置为 15 而不会影响答案。
- 类似地开始时间在 (15,30) 可以置为 30。
- ...

需要注意的是对于 (45, 60) 置为 0 的过程，需要将小时进位。

结束时间也是类似的，不再赘述，大家看代码即可。

接下来，我们计算结束时间和开始时间之间的分钟差 `span`，计算 `span` 拥有多少完成的 15 min 即可，也就是说可以用 `span` 整除 15 即可。

关键点

- 将开始时间和结束时间**规范到**标准时间

代码

- 语言支持: Python3

Python3 Code:

```
class Solution:
    def numberOfRounds(self, startTime: str, finishTime: str) -> int:
        sh, sm = map(int, startTime.split(":"))
        eh, em = map(int, finishTime.split(":"))
        d = 0
        if sh * 60 + sm > eh * 60 + em: d += 1
        if 0 < sm <= 15:
            sm = 15
        elif 15 < sm <= 30:
            sm = 30
        elif 30 < sm <= 45:
            sm = 45
        elif 45 < sm <= 60:
            sm = 0
            sh += 1
        if 0 <= em < 15:
            em = 0
        elif 15 <= em < 30:
            em = 15
        elif 30 <= em < 45:
            em = 30
        elif 45 <= em < 60:
            em = 45
        st = sh * 60 + sm
        et = eh * 60 + em
        if d == 1: et += 24 * 60
        return max(0, (et - st)) // 15
```

复杂度分析

- 时间复杂度: $O(1)$
- 空间复杂度: $O(1)$

此题解由 [力扣刷题插件](#) 自动生成。

力扣的小伙伴可以[关注我](#)，这样就会第一时间收到我的动态啦~

以上就是本文的全部内容了。大家对此有何看法，欢迎给我留言，我有时间都会一一查看回答。更多算法套路可以访问我的 LeetCode 题解仓库：<https://github.com/azl397985856/leetcode>。目前已经 40K star 啦。大家也可以关注我的公众号《力扣加加》带你啃下算法这块硬骨头。

关注公众号力扣加加，努力用清晰直白的语言还原解题思路，并且有大量图解，手把手教你识别套路，高效刷题。



欢迎长按关注



努力做西湖区
最好的算法题解

上一页

下一页



© 2020 lucifer. 保留所有权利