[1010. 总持续时间可被 60 整除的歌曲](https://leetcode-cn.com/problems/pairs-of-songs-with-total-durations-divisible-by-60/)

在歌曲列表中，第 i 首歌曲的持续时间为 time[i] 秒。

返回其总持续时间（以秒为单位）可被 60 整除的歌曲对的数量。形式上，我们希望索引的数字  i < j 且有 (time[i] + time[j]) % 60 == 0。

示例 1：

输入：[30,20,150,100,40]

输出：3

解释：这三对的总持续时间可被 60 整数：

(time[0] = 30, time[2] = 150): 总持续时间 180

(time[1] = 20, time[3] = 100): 总持续时间 120

(time[1] = 20, time[4] = 40): 总持续时间 60

示例 2：

输入：[60,60,60]

输出：3

解释：所有三对的总持续时间都是 120，可以被 60 整数。

提示：

1 <= time.length <= 60000

1 <= time[i] <= 500

解决方案1

双重for循环，遍历2次数组，依次计算每2首歌的时间和，再除以60，如果能整除，计数器+1，最终可以得到结果，时间复杂度为O(n^2)

解决方案2

构建一个长度为60的int数组arr，

遍历数组，依次计算每首歌除以60的余数，数组对应位置的计数器+1，最终可以获得余数为1秒，2秒，...，59秒的歌曲总和，可以得知，余数为1和余数为59的2首歌满足条件，

可以求得返回结果是arr[1] \* arr[59] + arr[2] \* arr[58] + ... + arr[29] \* arr[31]，

再处理0秒和30秒的数据

余数为0，任意2首歌都满足条件，所以需要加上 arr[0] \* (arr[0] - 1) / 2

余数为30，任意2首歌都满足条件，所以要加上arr[30] \* (arr[30] - 1) / 2

时间复杂度为O(n)

额外的空间int[60]