

题目：

在柠檬水摊上，每一杯柠檬水的售价为 5 美元。

顾客排队购买你的产品，（按账单 bills 支付的顺序）一次购买一杯。

每位顾客只买一杯柠檬水，然后向你付 5 美元、10 美元或 20 美元。你必须给每个顾客正确找零，也就是说净交易是每位顾客向你支付 5 美元。

注意，一开始你手头没有任何零钱。

如果你能给每位顾客正确找零，返回 true ，否则返回 false 。

示例 1：

```
输入：[5,5,5,10,20]
输出：true
解释：
前 3 位顾客那里，我们按顺序收取 3 张 5 美元的钞票。
第 4 位顾客那里，我们收取一张 10 美元的钞票，并返还 5 美元。
第 5 位顾客那里，我们找还一张 10 美元的钞票和一张 5 美元的钞票。
由于所有客户都得到了正确的找零，所以我们输出 true。
```

示例 2：

```
输入：[5,5,10]
输出：true
```

示例 3：

```
输入：[10,10]
输出：false
```

示例 4：

```
输入：[5,5,10,10,20]
输出：false
解释：
前 2 位顾客那里，我们按顺序收取 2 张 5 美元的钞票。
对于接下来的 2 位顾客，我们收取一张 10 美元的钞票，然后返还 5 美元。
对于最后一位顾客，我们无法退回 15 美元，因为我们现在只有两张 10 美元的钞票。
由于不是每位顾客都得到了正确的找零，所以答案是 false。
```

提示：

- 0 <= bills.length <= 10000
- bills[i] 不是 5 就是 10 或是 20

思路：

遍历bills，记录 5和10 的个数

如果是5， 不需要找零，就把5 的个数+1

如果是10， 如果能有15的话，那么就可以找零了

有5：就把5的个数-1， 10的个数+1

没有，返回False

如果是20 ， 如果能有15的话，那么就可以找零了

那么15有两种可能：

1 有10，有5，10，5的个数分别 -1

2 有3个5，就把5的个数-3

解题：

```
1 def lemonadeChange(self, bills):
2     """
3     :type bills: List[int]
4     :rtype: bool
5     """
6     five = ten = 0
7     for bill in bills:
8         if bill == 5:
9             five += 1
10        elif bill == 10:
11            if not five :
12                return False
13            five -= 1
14            ten +=1
15        else:
16            if ten and five:
17                ten -= 1
18                five -= 1
19            elif five >=3:
20                five -= 3
21            else:
22                return False
23    return True
```