题目难度：EASY

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：贪心

初始化一个两空间初值0，记录5元和10元个数

遍历数组，每次先把遇到的是5元还是10元累加金进数组

接下来如果当前是10元，那么判断5元个数大于0就把数组值减一

如果是20元先看有没有10元，有对应数组减一，

再看有没有5元，有对应数组减一，否则返回FALSE

否则判断有没有三个5元，有对应数组值减三，否则返回FALSE（WA点：缺少此处判断）

最后数组遍历结束返回TRUE即可

时间击败79%（时间不错）

空间击败42%（空间一般）

上代码：

class Solution {

public:

int curr[2] = {0, 0};

bool lemonadeChange(vector<int>& bills) {

for(int i = 0; i < bills.size(); i++) {

if(bills[i] != 20) curr[bills[i] / 5 - 1]++;

if(bills[i] == 10) {

if(curr[0] == 0) return false;

curr[0]--;

}

if(bills[i] == 20){

if(curr[1] > 0) {

curr[1]--;

if(curr[0] == 0) return false;

else curr[0]--;

} else {

if(curr[0] < 3) return false;

else curr[0] -= 3;

}

}

}

return true;

}

};