860. Lemonade Change

At a lemonade stand, each lemonade costs `$5`. Customers are standing in a queue to buy from you and order one at a time (in the order specified by bills). Each customer will only buy one lemonade and pay with either a `$5`, `$10`, or `$20` bill. You must provide the correct change to each customer so that the net transaction is that the customer pays `$5`.

Note that you do not have any change in hand at first.

Given an integer array `bills` where `bills[i]` is the bill the `i`th customer pays, return `true` *if you can provide every customer with the correct change, or* `false` *otherwise*.

Example 1:

```
Input: bills = [5,5,5,10,20]
Output: true
Explanation:
From the first 3 customers, we collect three $5 bills in order.
From the fourth customer, we collect a $10 bill and give back a $5.
From the fifth customer, we give a $10 bill and a $5 bill.
Since all customers got correct change, we output true.
```

Example 2:

```
Input: bills = [5,5,10,10,20]
Output: false
Explanation:
From the first two customers in order, we collect two $5 bills.
For the next two customers in order, we collect a $10 bill and give back a $5 bill.
For the last customer, we can not give the change of $15 back because we only have
two $10 bills.
Since not every customer received the correct change, the answer is false.
```

Constraints:

- `1 <= bills.length <= 10`5
- `bills[i]` is either `5`, `10`, or `20`.

860. Lemonade Change

```
/*
    Greedy
    Time complexity: o(n)
    Space complexity: O(1)
*/
class Solution {
public:
    bool lemonadeChange(vector<int>& bills) {
        if(bills[0]!=5) return false;

        int five=1;
        int ten=0;

        int n=bills.size();
        for(int i=1;i<n;++i){
            int bill=bills[i];
            if(bill==5) five++;
            else if(bill==10) ten++;
            int change=bill-5;
            if(change==5){
                if(five<1) return false;
                five--;
            }
            if(change==15){
                if(five<3 && (ten<1 || five<1)) return false;

                if(ten>=1 && five>=1) ten--,five--;
                else if(five>=3) five-=3;
            }
        }
        return true;
    }
};
```