一、题目

在一条环路上有 N 个加油站，其中第 i 个加油站有汽油 gas[i] 升。
你有一辆油箱容量无限的的汽车，从第 i 个加油站开往第 i+1 个加油站需要消耗汽油 cost[i] 升。你从其中的一个加油站出发，开始时油箱为空。
如果你可以绕环路行驶一周，则返回出发时加油站的编号，否则返回 -1。

说明:
如果题目有解，该答案即为唯一答案。
输入数组均为非空数组，且长度相同。
输入数组中的元素均为非负数。

示例 1:
输入:
```
gas  = [1,2,3,4,5]
cost = [3,4,5,1,2]
```

输出: 3

解释:
从 3 号加油站(索引为 3 处)出发，可获得 4 升汽油。此时油箱有 = 0 + 4 = 4 升汽油
开往 4 号加油站，此时油箱有 4 - 1 + 5 = 8 升汽油
开往 0 号加油站，此时油箱有 8 - 2 + 1 = 7 升汽油
开往 1 号加油站，此时油箱有 7 - 3 + 2 = 6 升汽油
开往 2 号加油站，此时油箱有 6 - 4 + 3 = 5 升汽油
开往 3 号加油站，你需要消耗 5 升汽油，正好足够你返回到 3 号加油站。
因此，3 可为起始索引。

示例 2:
输入:
```
gas  = [2,3,4]
cost = [3,4,3]
```

输出: -1

解释:
你不能从 0 号或 1 号加油站出发，因为没有足够的汽油可以让你行驶到下一个加油站。
我们从 2 号加油站出发，可以获得 4 升汽油。 此时油箱有 = 0 + 4 = 4 升汽油
开往 0 号加油站，此时油箱有 4 - 3 + 2 = 3 升汽油
开往 1 号加油站，此时油箱有 3 - 3 + 3 = 3 升汽油
你无法返回 2 号加油站，因为返程需要消耗 4 升汽油，但是你的油箱只有 3 升汽油。
因此，无论怎样，你都不可能绕环路行驶一周。

二、思路

> 前提知识：如果一个数组的总和非负，那么一定可以找到一个起始位置，从他开始绕数组一圈，累加和一直保持非负。

1、因此只要计算出一圈后加油与用油最后的总和是否大于0即可判断是否有解。gas[i] >= cost[i]即可走向下一站

2、对于找到起始位置，可以遍历数组，假设从i开始前进，到达j的时候没油了，那么我们下一步不应该从i+1开始遍历，而是应该直接从j+1开始遍历。

3、因为如果i到j的剩余油量小于0，而i显然油量大于0，那么从i+1到j就必定更小，同理，i+2,i+3也不用考虑，所以就应该直接从j+1开始继续遍历，并保存之前欠缺的油量总和。

三、代码

```
public static int canCompleteCircuit(int[] gas, int[] cost) {
        int sum = 0;       //油箱里的油  可能为负数
        int total = 0;
        int start = 0;  //记录从哪个位置开始出发
        for (int i = 0; i < gas.length; i++) {
            sum += gas[i] - cost[i];       //到达下一个位置剩下的油  可能为负数
            if (sum < 0) {  //如果为负数  表示从start到i的位置 都不能作为开始位置
                start = i + 1;
                total += sum; //total为到达i时总共要付出的油，total一直为负数
                sum = 0;       //从下一位置开始  所以sum置为0
            }
        }
        total = total + sum; //total为负数  即是最终累计剩下的sum减去要付出的总和total
        return total < 0 ? -1 : start;//total小于0就说明总gas小于总cost
    }
```

> **注：可以通过复制以上代码到 https://tool.lu/coderunner/ 验证代码**

四、复杂度

- 时间复杂度：O(n)
- 空间复杂度：O(1)