

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，无论如何，你都不可能绕环路行驶一周。

思路，这题其实就是遍历数组，若是消耗的值小于加油的值，则不可能环绕，只要大于或者等于，那么其中必能从一个数开始，相加都大于0。那么动态判断相加值到最后即可，若是一个数能从该数开始，相加到结尾，值依旧为正，那么他能从结尾到开头相加为正（因为你一个环形，和为正，

那么这个环就分左右两端，左端为正，右端必比左端小的负数，两端为从0到你这值为负，你这到结尾值为正)

代码

```
class Solution {
public:
    int canCompleteCircuit(vector<int>& gas, vector<int>& cost) {
        vector<int> nums(gas.size(), 0);
        int num=0;
        for(int i=0;i<gas.size();i++){
            nums[i]=gas[i]-cost[i];
            num+=nums[i];
        }
        if(num<0){
            return -1;
        }
        int wz=0;
        for(int i=0,num=0;i<nums.size();i++){
            if(num==0){
                if(nums[i]>0){
                    wz=i;
                    num=nums[i];
                }
            }else{
                num+=nums[i];
                if(num<0){
                    num=0;
                }
            }
        }
        return wz;
    }
};
```