

0134. 加油站

👤 [ITCharge](#) ⌚ 大约 2 分钟

- 标签：贪心、数组
- 难度：中等

题目链接

- [0134. 加油站 - 力扣](#)

题目大意

一条环路上有 N 个加油站，第 i 个加油站有 $gas[i]$ 升汽油。

现在有一辆油箱无限容量的汽车，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。如果汽车上携带的有两不够 $cost[i]$ ，则无法从第 i 个加油站开往第 $i + 1$ 个加油站。

现在从其中一个加油站开始出发，且 时油箱为空。如果能绕环路行驶一周，则返回出发时加油站的编号，否则返回 -1。

解题思路

1. 暴力求解

分别考虑从第 0 个点、第 1 个点、...、第 i 个点出发，能否回到第 0 个点、第 1 个点、...、第 i 个点。

2. 贪心算法

- 如果加油站提供的油总和大于等于消耗的汽油量，则必定可以绕环路行驶一周
- 假设先不考虑油量为负的情况，我们从「第 0 个加油站」出发，环行一周。记录下汽油量 $gas[i]$ 和 $cost[i]$ 差值总和 sum_diff ，同时记录下油箱剩余油量的最小值 min_sum 。
- 如果差值总和 $sum_diff < 0$ ，则无论如何都不能环行一周。油不够啊，亲！！
- 如果 $min_sum \geq 0$ ，则行驶过程中油箱始终有油，则可以从 0 个加油站出发环行一周。
- 如果 $min_sum < 0$ ，则说明行驶过程中油箱油不够了，那么考虑更换开始的起点。

- 从右至左遍历，计算汽油量 $gas[i]$ 和 $cost[i]$ 差值，看哪个加油站能将 min_sum 填平。如果最终达到 $min_sum \geq 0$ ，则说明从该点开始出发，油箱中的油始终不为空，则返回该点下标。
- 如果找不到最返回 -1。

代码

```
class Solution:
    def canCompleteCircuit(self, gas: List[int], cost: List[int]) -> int:
        sum_diff, min_sum = 0, float('inf')
        for i in range(len(gas)):
            sum_diff += gas[i] - cost[i]
            min_sum = min(min_sum, sum_diff)

        if sum_diff < 0:
            return -1

        if min_sum >= 0:
            return 0

        for i in range(len(gas)-1, -1):
            min_sum += gas[i] - cost[i]
            if min_sum >= 0:
                return i
        return -1
```

py

参考链接

- [贪心算法/前缀和 - 加油站 - 力扣 \(LeetCode\)](#)