

LeetCode 134

Problem Description : [Gas Station](#)/加油站

Language: C++, Python3

Label: Greedy/贪心, Brain Teaser /脑筋急转弯

Solution:

显然模拟一个起点是否能够达到终点需要 $O(n)$ 的时间复杂度，那么枚举每个点的起点就需要 $O(n^2)$ 的复杂度，题目中虽然没有说明 n 的规模，但我们肯定希望用更优的解法。考虑这样一个情况，假设有加油站 X_0 ，我们从 X_0 开始枚举，遇到 X_N 时汽油不够，此时停止循环，得到了 $X_0, X_1, X_2, \dots, X_N$ 的队列，那么我们可以保证 X_0 开始抵达任意一个 $X_i (i < N)$ 均有 $ExtraGas \geq 0$ ，然后考虑起始点为 X_i 时到 X_N 的路径，此时 $ExtraGas = 0$ ，那么相比于 X_0 到 X_N ，失去了额外油量的加持，也一定无法抵达。即若 $ExtraGas + gas[i] < cost$ ，那么 $gas[i] < cost$ 必定成立。

这就意味着对于 $X_i (0 \leq i \leq N)$ 均无法完成一周的旅程，我们的下一次枚举点之间变更为 X_{N+1} ，基于这样的思想，实际上只要遍历一遍数组，时间复杂度为 $O(n)$

C++ Source Code

```
class Solution
{
public:
    int canCompleteCircuit(vector<int>& gas, vector<int>& cost)
    {
        int n = gas.size();
        int index = 0; //起点下标
        while(index < n)
        {
            int ExtraGas = 0, i;
            for(i = 0; i < n; i++) //枚举n个点
            {
                int j = (i+index)%n; //类似循环队列
                ExtraGas += gas[j]-cost[j];
                if(ExtraGas < 0) break;
            }
            if(i == n) return index;
            else index += i+1; //getNext
        }
        return -1; //没有符合要求的点
    }
};
```

Python3 Source Code

```
class Solution:
    def canCompleteCircuit(self, gas: List[int], cost: List[int]) -> int:
        n = len(gas)
        index = 0
        while index < n:
            ExtraGas = i = 0
            while i < n:
                j = (i+index)%n
                ExtraGas += gas[j]-cost[j]
                if ExtraGas < 0:
                    break
                i += 1

            if i == n:
                return index
            index += i+1

        return -1
```

Author@[Kuroko](#)

GitHub@[SuperKuroko](#)

LeetCode@[kuroko177](#)

Last Modified: 2020-11-18 12:55