원형으로 경로가 연결된 주유소 목록이 있다. 각주유소는 gas[i]만큼의 기름을 같고 있으며, 다음 주유소로 이동하는데 cost[i]가 필요하다. 기름이 부족하면 이동할 수 없다고 할 때 모든 주유소를 방문할 수 있는 출발점의 인덱스를 출력하라.
출발점이 존재하지 않을 경우 -1을 리턴하며, 출발점은 유일하다.

Input: gas = [1,2,3,4,5], cost = [3,4,5,1,2]
Output: 3
Explanation:
Start at station 3 (index 3) and fill up with 4 unit of gas. Your tank = 0 + 4 = 4
Travel to station 4. Your tank = 4 - 1 + 5 = 8
Travel to station 0. Your tank = 8 - 2 + 1 = 7
Travel to station 1. Your tank = 7 - 3 + 2 = 6
Travel to station 2. Your tank = 6 - 4 + 3 = 5
Travel to station 3. The cost is 5. Your gas is just enough to travel back to station 3.
Therefore, return 3 as the starting index.

1.그리디

```
class Solution:
    def canCompleteCircuit(self, gas: List[int], cost: List[int]) -> int:
        candi = [i for i in range(len(gas)) if gas[i] >= cost[i]]
        for c in candi:
            flag = True
            i = (c + 1) % len(gas)
            left = gas[c] - cost[c]
            while i != c:
                left += gas[i]
                left -= cost[i]
                i = (i + 1) % len(gas)
                if left < 0:
                    flag = False
                    break
            if flag:
                return c
        return -1
```

```
class Solution:
    def canCompleteCircuit(self, gas: List[int], cost: List[int]) -> int:
        for start in range(len(gas)):
            fuel = 0
            for i in range(start, len(gas) + start):
                index = i % len(gas)
                can_travel = True
                if gas[index] + fuel < cost[index]:
                    can_travel = False
                    break
                else:
                    fuel += gas[index] - cost[index]
            if can_travel:
                return start
        return -1
```

2.귀류법

```
class Solution:
    def canCompleteCircuit(self, gas: List[int], cost: List[int]) -> int:
        if sum(gas) < sum(cost):
            return -1
        start, fuel = 0, 0
        for i in range(len(gas)):
            if gas[i] + fuel < cost[i]:
                start = i + 1
                fuel = 0
            else:
                fuel += gas[i] - cost[i]
        return start
```