Problem # 134 : Gas Station

https://leetcode.com/problems/gas-station/

My Solution :

<https://leetcode.com/problems/gas-station/discuss/861731/Simple-Python-3-Solution-with-selective-starting-points>

Simple Python 3 Solution with selective starting points

The brute force method would be to have each index as the starting position and see if there is a complete circuit. In the worst case, if there is no circuit the solution would be O(n^2).

I analyzed the problem and found that the index for the starting point has to be either the index(es) corresponding to the minimum value from cost array or maximum value from gas array.

1. Find the minimum cost from the cost array.
2. Find the maximum gas from the gas array.
3. Get the starting position list corresponding to indexes having either minimum cost or maximum gas.
4. Check complete circuit only for these indexes present in the starting position list. If a full circuit is detected, return the starting position.
5. However, if we have checked all starting position and none of them have a full circuit, then return -1.

class Solution:

def canCompleteCircuit(self, gas: List[int], cost: List[int]) -> int:

n = len(gas)

min\_cost = min(cost)

max\_gas = max(gas)

start\_pos\_list = []

for i, c in enumerate(cost):

if c == min\_cost or gas[i] == max\_gas:

start\_pos\_list.append(i)

for start\_pos in start\_pos\_list:

isComplete = True

quantity = 0

pos = start\_pos

for i in range(n):

quantity += gas[pos] - cost[pos]

print("i = ", i, "pos = ", pos, "quantity = ", quantity)

pos = (pos + 1)% n

if quantity < 0:

isComplete = False

break

if isComplete:

return(start\_pos)

return(-1)