

Team id	IBM-Project-27329-1660054173
date	19/12/2022
Project name	Car resale value prediction Using Machine learning

The Car Fueling Problem

Solving this algorithm problem in Python



Problem Introduction

You are going to travel to another city that is located d miles away from your home city. Your car can travel.

At most m miles on a full tank and you start with a full tank. Along your way, there are gas stations at

Distances $\text{stop}_1, \text{stop}_2, \dots, \text{stop}_n$ from your home city. What is the minimum number of refills needed?

Problem Description

Input Format. The first line contains an integer d . The second line contains an integer m . The third line

Specifies an integer n . Finally, the last line contains integers $\text{stop}_1, \text{stop}_2, \dots, \text{stop}_n$.

Output Format

Assuming that the distance between the cities is d miles, a car can travel at most m miles

On a full tank, and there are gas stations at distances $\text{stop}_1, \text{stop}_2, \dots, \text{stop}_n$ along the way, output the

Minimum number of refills needed. Assume that the car starts with a full tank. If it is not possible to

Reach the destination, output -1 .

Constraints

$1 \leq d \leq 105$. $1 \leq m \leq 400$. $1 \leq n \leq 300$. $0 < \text{stop}_1 < \text{stop}_2 < \dots < \text{stop}_n < d$.

Sample 1

Input:

950 400 4200 375 550 750

Output:

2

The distance between the cities is 950, the car can travel at most 400 miles on a full tank. It suffices

to make two refills: at points 375 and 750. This is the minimum number of refills as with a single refill

One would only be able to travel at most 800 miles.

Sample 2

Input:

10341 2 5 9

Output:

-1

One cannot reach the gas station at point 9 as the previous gas station is too far away.

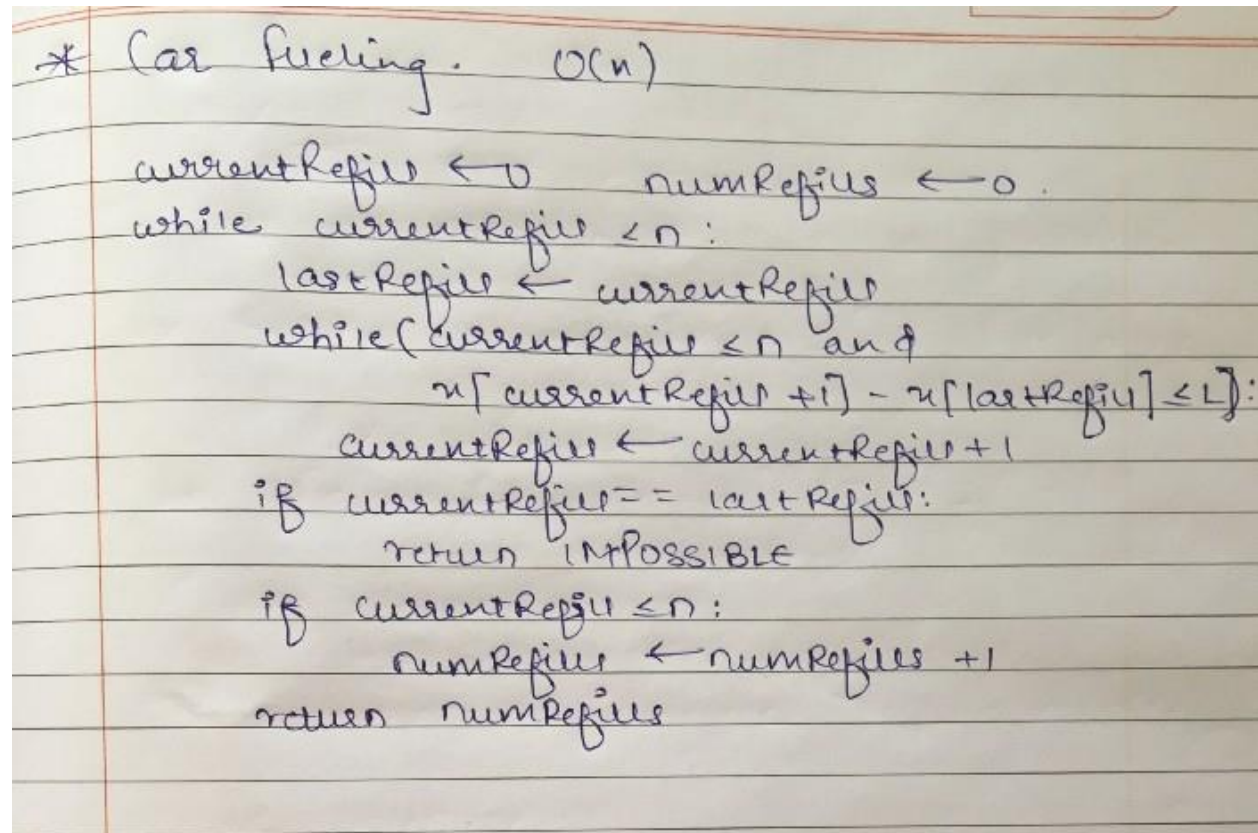
Greedy Strategy

1. Make a greedy choice!
2. Reduce to a smaller problem
3. Iterate

A greedy choice is a safe move if there is an optimal solution consistent with the first move:

4. Refill at the closest gas station
5. Refill at the farthest reachable gas station
6. Go until the fuel finishes up!

Implementation



```
* Car fueling.  $O(n)$ 

currentRefill  $\leftarrow 0$     numRefills  $\leftarrow 0$ 
while currentRefill  $< n$ :
    lastRefill  $\leftarrow$  currentRefill
    while (currentRefill  $< n$  and
            $x[\text{currentRefill} + 1] - x[\text{lastRefill}] \leq L$ ):
        currentRefill  $\leftarrow$  currentRefill + 1
    if currentRefill == lastRefill:
        return IMPOSSIBLE
    if currentRefill  $\leq n$ :
        numRefills  $\leftarrow$  numRefills + 1
    return numRefills
```

the algorithm

Here, n is the total number of stops, x is an array with distance d i.e. $x[d]$, currentRefill , lastRefills are nothing but indexes. Also, numRefills is the minimum number of fills.

Code

```
def car_fueling(dist,miles,n,gas_stations):

    num_refill, curr_refill, limit = 0,0,miles
    while limit < dist:
        # While the destination cannot be reached with current fuel
        if curr_refill >= n or gas_stations[curr_refill] > limit:
            # Cannot reach the destination nor the next gas station
            return -1

        # Find the furthest gas station we can reach
        while curr_refill < n-1 and gas_stations[curr_refill+1] <= limit:
            curr_refill += 1
```

```
    num_refill += 1 # Stop to tank
    limit = gas_stations[curr_refill] + miles # Fill up the tank
    curr_refill += 1

return num_refill
```