**Corrected Scenario-Based Coding Problem: Minimum Number of Stops**

**Problem Statement:**

A car travels from a starting position to a destination which is target miles east of the starting position. There are gas stations along the way. The gas stations are represented as an array stations where stations[i] = [positioni, fueli] indicates that the ith gas station is positioni miles east of the starting position and has fueli liters of gas. The car starts with an infinite tank of gas, which initially has startFuel liters of fuel in it. It uses one liter of gas per one mile that it drives. When the car reaches a gas station, it may stop and refuel, transferring all the gas from the station into the car. Note that if the car reaches a gas station with 0 fuel left, the car can still refuel there. If the car reaches the destination with 0 fuel left, it is still considered to have arrived.

**Input Format:**

* First line contains target
* Second line contains startFuel
* Third line contains an integer n representing the number of gas stations
* Next n lines each contain two integers positioni and fueli

**Output Format:**

* Return the minimum number of refueling stops the car must make in order to reach its destination. If it cannot reach the destination, return -1.

**Constraints:**

* 1 <= target, startFuel <= 10^9
* 0 <= stations.length <= 500
* 1 <= positioni < positioni+1 < target
* 1 <= fueli < 10^9

**Sample Input:**

100

10

4

10 60

20 30

30 30

60 40

**Sample Output:**

2

**Explanation:**

We start with 10 liters of fuel.

1. We drive to position 10, expending 10 liters of fuel. We refuel from 0 liters to 60 liters of gas.
2. Then, we drive from position 10 to position 60 (expending 50 liters of fuel), and refuel from 10 liters to 50 liters of gas. We then drive to and reach the target. We made 2 refueling stops along the way, so we return 2.

**Solution:**

python

import heapq

def minRefuelStops(target, startFuel, stations):

# Convert the stations array to a format suitable for heap operations

stations.append([target, 0])

max\_heap = []

stops, prev, fuel = 0, 0, startFuel

for location, capacity in stations:

fuel -= (location - prev)

while max\_heap and fuel < 0: # Must refuel in order to reach this location

fuel += -heapq.heappop(max\_heap)

stops += 1

if fuel < 0:

return -1

heapq.heappush(max\_heap, -capacity)

prev = location

return stops

# Reading input

target = int(input())

startFuel = int(input())

n = int(input())

stations = [list(map(int, input().split())) for \_ in range(n)]

# Solving the problem

result = minRefuelStops(target, startFuel, stations)

print(result)

**Test Cases:**

**Test Case 1:**

Input:

100

10

4

10 60

20 30

30 30

60 40

Output:

2

**Test Case 2:**

Input:

200

50

3

50 50

100 50

150 50

Output:

3

**Test Case 3:**

Input:

1000

100

4

200 100

400 100

600 100

800 100

Output:

-1

**Test Case 4:**

Input:

100

50

1

50 25

Output:

-1

**Test Case 5:**

Input:

100

100

0

Output:

0

This problem tests the ability to efficiently simulate the refueling process using a max heap to always refuel with the largest available fuel capacity when necessary. The solution ensures the car can reach the destination with the minimum number of stops or determines if it's impossible.