

truck tour hackerrank solution

Truck tour hackerrank solution is a popular problem found on the HackerRank platform, which challenges participants to analyze and solve a logistical problem involving a circular route for trucks. This problem not only tests your coding skills but also your ability to think algorithmically. In this article, we will delve into the details of the Truck Tour problem, explore various approaches to arrive at the solution, and provide an efficient coding solution that can help you excel in coding interviews and competitions.

Understanding the Truck Tour Problem

The Truck Tour problem involves a circular route where multiple petrol stations are positioned around a track. Each station has a certain amount of petrol and a distance to the next station. The goal is to determine if there exists a starting petrol station such that the truck can complete the circuit without running out of petrol.

Problem Statement

You are given:

- An integer `n` representing the number of petrol stations.
- An array of integers `petrol[]` where `petrol[i]` is the amount of petrol at the i -th station.
- An array of integers `distance[]` where `distance[i]` is the distance from the i -th station to the $(i+1)$ -th station.

You need to find the starting station index from where the truck can complete the full circular tour. If it is impossible to complete the tour, you should return -1.

Key Insights for the Solution

To solve the Truck Tour problem effectively, it's essential to consider the following insights:

1. **Total Petrol vs. Total Distance:** If the total amount of petrol available is less than the total distance required to complete the tour, it's impossible to finish the circuit. So, at the outset, we can quickly eliminate those scenarios.
2. **Greedy Approach:** This problem can be solved using a greedy algorithm. If

the truck fails to reach a station, the previous stations can be eliminated as potential starting points since they would not have enough petrol to reach the next station.

3. Single Pass: The goal is to find the solution in a single pass through the data. This will help achieve an optimal time complexity of $O(n)$.

Steps to Solve the Problem

To implement the solution, follow these steps:

1. Calculate the total petrol and total distance.
2. If the total petrol is less than the total distance, return -1.
3. Initialize variables to track the starting index and the current petrol balance.
4. Traverse through the petrol stations:
 - Update the petrol balance.
 - Check if the balance goes negative; if so, reset the starting station to the next index.
5. Return the starting station index if a valid starting point is found.

Implementation of the Solution

Here's a Python implementation of the Truck Tour hackerrank solution:

```
```python
def truck_tour(petrol, distance):
 total_petrol = 0
 total_distance = 0
 current_petrol = 0
 start_index = 0

 for i in range(len(petrol)):
 total_petrol += petrol[i]
 total_distance += distance[i]
 current_petrol += petrol[i] - distance[i]
```

If we can't reach the next station, we must start from the next station

```
if current_petrol < 0:
 start_index = i + 1
 current_petrol = 0
```

We can only complete the tour if total petrol is greater than or equal to total distance

```
if total_petrol < total_distance:
 return -1
```

```
return start_index
'''
```

## Example Usage

Let's take a look at an example to understand how our solution works:

```
```python
petrol = [4, 6, 7, 4]
distance = [6, 5, 3, 5]

start_station = truck_tour(petrol, distance)
print(f"The starting station index is: {start_station}")
```
```

In this case, the output will be the index of the starting petrol station from which the truck can complete the circular route.

## Testing the Solution

When implementing a solution, it's crucial to test it against various scenarios to ensure its robustness. Here are some test cases:

1. Basic Test Case: Where the total petrol is equal to the total distance.
2. Edge Case: All petrol stations provide more petrol than required to reach the next station.
3. Impossible Tour: Where total petrol is less than total distance.

```
```python
Test case 1
assert truck_tour([1, 2, 3, 4], [2, 3, 4, 3]) == 3

Test case 2
assert truck_tour([2, 3, 4], [3, 4, 3]) == 1

Test case 3
assert truck_tour([1, 2, 3, 4], [4, 5, 6, 7]) == -1
```
```

## Conclusion

**Truck tour hackerrank solution** is an excellent problem that combines algorithmic thinking with practical applications in logistics. By understanding the core concepts and applying a greedy approach, you can solve

this problem efficiently. Whether you're preparing for coding interviews or competing in coding challenges, mastering this problem will significantly enhance your problem-solving skills. Practice with various test cases to solidify your understanding and ensure that you can tackle similar problems with confidence.

## **Frequently Asked Questions**

### **What is the Truck Tour problem on HackerRank?**

The Truck Tour problem requires finding the starting petrol pump index from which a truck can complete a circular tour without running out of petrol, given the petrol and distance between consecutive pumps.

### **What are the key inputs for the Truck Tour problem?**

The key inputs are two arrays: one representing the amount of petrol available at each petrol pump and another representing the distance to the next pump.

### **What is the expected output for the Truck Tour problem?**

The expected output is the index of the petrol pump from which the truck can start its journey and complete the tour successfully.

### **What is the time complexity of the optimal solution for the Truck Tour problem?**

The optimal solution for the Truck Tour problem has a time complexity of  $O(n)$ , where  $n$  is the number of petrol pumps.

### **Could you provide a brief approach to solving the Truck Tour problem?**

A common approach is to iterate through the petrol pumps while maintaining two variables: total petrol and total distance. If at any point total petrol becomes negative, reset the starting index to the next pump and reset the totals.

### **What common mistakes should be avoided when solving the Truck Tour problem?**

Common mistakes include not correctly updating the total petrol and distance or failing to reset the starting index when the current route fails.

## Is there a specific function signature for the Truck Tour problem on HackerRank?

Yes, the function typically follows the signature: `'int truckTour(vector<pair<int, int>> petrolPumps)'`, where each pair contains the petrol and distance values.

## What are some edge cases to consider in the Truck Tour problem?

Edge cases include having only one petrol pump, all pumps providing insufficient petrol, or all pumps providing excess petrol.

## Are there any variations of the Truck Tour problem?

Variations can include constraints on the amount of petrol or introducing obstacles that affect the distance between pumps.

## Where can I find discussions or solutions for the Truck Tour problem?

You can find discussions and solutions in the HackerRank community forums, GitHub repositories, or competitive programming blogs.

## [Truck Tour Hackerrank Solution](#)

Truck Tour Hackerrank Solution

## Related Articles

- [transaction processing concepts and techniques](#)
- [unit 5 geography challenge answer key](#)
- [uc davis olive oil study](#)

[Back to Home](#)