

Greedy Algorithm Homework

Auiannce Euwing

October 2025

1 Part A

Correctness Explanation My algorithm works because at each step it chooses the unvisited neighboring city that gives the company the most immediate profit, which is the delivery fee minus the travel cost to get there. By always taking the best available option, the truck builds a route that never wastes money on a worse choice when a better one is available. The loop continues until there are no new cities to visit, so every customer that can be reached is visited once. Finally, the algorithm subtracts the travel cost to return to the depot, giving the correct total profit. The algorithm correctly follows the greedy rule by always making the locally best profit decision.

2 Part B

Correctness Explanation This algorithm is correct because it always makes the best possible local decision according to the driver's goal. At each step, the driver looks at all unvisited neighboring customers and calculates how much money could be earned from going there next. The calculation correctly includes all parts of the driver's earnings: delivery fee, estimated tip, and travel cost. By always choosing the customer that gives the highest immediate earnings, the driver never misses a better option that is available at that moment. The algorithm also keeps an accurate total by adding every fee and tip and subtracting each travel cost, including the return trip to the depot. Because every customer is visited only once and the route always follows valid roads, the algorithm produces a valid route and a correct total earnings value.

3 Part C

Goal. The goal is to think about the responsibility of the algorithm design and to reflect ethical considerations in the greedy decision process. Even though greedy algorithms often make efficient, profit-maximizing choices, they can overlook human factors such as driver safety, fatigue, or fairness among customers. To address this, I chose to modify the driver's greedy algorithm from Part B to consider fatigue.

Ethical Rule Implemented. In the original algorithm, the driver always picked the next customer that maximized earnings, which included tips. To include a fatigue rule, I added a constraint. If the last trip was long (greater than a certain distance), then the next trip must be short (below a short distance limit), if such a nearby customer exists. This rule represents a real-world adjustment that helps reduce how tired a driver gets from taking too many long, back-to-back drives.

Results and comparison. I ran the code and seem to get the same route, with the same amount of profit for both part B and part C. This means that when a driver prioritizes tips, they are also selecting routes that would lead to less exhaustion.