Assignment: Design and Analysis of Algorithms

Instructions:

This assignment is designed to help you apply the concepts learned in the course to real-world problems relevant to current industry needs. Each problem should be solved with a clear and thorough explanation of your approach and reasoning. Please ensure that your solutions are well-documented and that you provide both pseudocode and actual code (where applicable).

Due Date: July 1 2024

Problem 1: Optimizing Delivery Routes (Case Study)

Scenario: You are working for a logistics company that wants to optimize its delivery routes to minimize fuel consumption and delivery time. The company operates in a city with a complex road network.

Tasks:

1. Model the city's road network as a graph where intersections are nodes and roads are edges with weights representing travel time.
2. Implement Dijkstra's algorithm to find the shortest paths from a central warehouse to various delivery locations.
3. Analyze the efficiency of your algorithm and discuss any potential improvements or alternative algorithms that could be used.

Deliverables:

- Graph model of the city's road network.
- Pseudocode and implementation of Dijkstra's algorithm.
- Analysis of the algorithm's efficiency and potential improvements.

Reasoning: Explain why Dijkstra's algorithm is suitable for this problem. Discuss any assumptions made (e.g., non-negative weights) and how different road conditions (e.g., traffic, road closures) could affect your solution.

Problem 2: Dynamic Pricing Algorithm for E-commerce

Scenario: An e-commerce company wants to implement a dynamic pricing algorithm to adjust the prices of products in real-time based on demand and competitor prices.

Tasks:

1. Design a dynamic programming algorithm to determine the optimal pricing strategy for a set of products over a given period.
2. Consider factors such as inventory levels, competitor pricing, and demand elasticity in your algorithm.

3. Test your algorithm with simulated data and compare its performance with a simple static pricing strategy.

Deliverables:

- Pseudocode and implementation of the dynamic pricing algorithm.
- Simulation results comparing dynamic and static pricing strategies.
- Analysis of the benefits and drawbacks of dynamic pricing.

Reasoning: Justify the use of dynamic programming for this problem. Explain how you incorporated different factors into your algorithm and discuss any challenges faced during implementation.

Problem 3: Social Network Analysis (Case Study)

Scenario: A social media company wants to identify influential users within its network to target for marketing campaigns.

Tasks:

1. Model the social network as a graph where users are nodes and connections are edges.
2. Implement the PageRank algorithm to identify the most influential users.
3. Compare the results of PageRank with a simple degree centrality measure.

Deliverables:

- Graph model of the social network.
- Pseudocode and implementation of the PageRank algorithm.
- Comparison of PageRank and degree centrality results.

Reasoning: Discuss why PageRank is an effective measure for identifying influential users. Explain the differences between PageRank and degree centrality and why one might be preferred over the other in different scenarios.

Problem 4: Fraud Detection in Financial Transactions

Scenario: A financial institution wants to develop an algorithm to detect fraudulent transactions in real-time.

Tasks:

1. Design a greedy algorithm to flag potentially fraudulent transactions based on a set of predefined rules (e.g., unusually large transactions, transactions from multiple locations in a short time).
2. Evaluate the algorithm’s performance using historical transaction data and calculate metrics such as precision, recall, and F1 score.
3. Suggest and implement potential improvements to the algorithm.

Deliverables:

- Pseudocode and implementation of the fraud detection algorithm.
- Performance evaluation using historical data.
- Suggestions and implementation of improvements.

Reasoning: Explain why a greedy algorithm is suitable for real-time fraud detection. Discuss the trade-offs between speed and accuracy and how your algorithm addresses them.

Problem 5: Real-Time Traffic Management System

Scenario: A city's traffic management department wants to develop a system to manage traffic lights in real-time to reduce congestion.

Tasks:

1. Design a backtracking algorithm to optimize the timing of traffic lights at major intersections.
2. Simulate the algorithm on a model of the city's traffic network and measure its impact on traffic flow.
3. Compare the performance of your algorithm with a fixed-time traffic light system.

Deliverables:

- Pseudocode and implementation of the traffic light optimization algorithm.
- Simulation results and performance analysis.
- Comparison with a fixed-time traffic light system.

Reasoning: Justify the use of backtracking for this problem. Discuss the complexities involved in real-time traffic management and how your algorithm addresses them.

Submission Guidelines:

- Submit your assignment as a single PDF document containing all written explanations, pseudocode, and analysis.
- Include separate files for code implementations, with clear instructions on how to run them.
- Ensure that all graphs, charts, and simulation results are clearly labeled and easy to interpret.

Evaluation Criteria:

- **Correctness and Efficiency:** Solutions should be correct and optimized for efficiency.
- **Clarity and Documentation:** Solutions should be clearly explained and well-documented.
- **Reasoning and Justification:** Provide strong reasoning and justification for your approach and solutions.

Good luck, and remember to approach each problem methodically and thoughtfully!