ALGORITHM & DATA STRUCTURES

*Final Project*

## Introduction

Welcome to the final project!

The final project aims at applying the concepts you have acquired during course to a real-world problem.

You will identify a **real-world problem,** determine and justify the **choice of one or more data structures**, **implement** a solution, **analyze your solution** in term of complexity, and finally **document the process** thoroughly.

Projects will be performed in teams of 2 students.

Good luck everyone!

## Agenda

* Week 7: **Proposal Submission**

*Teams will submit a one-page proposal outlining the problem, ADT(s) to be used, and the planned implementation strategy.*

* Week 10: **Project Defense and Final Submission**

*Teams will deliver a final report, present their project, and defend their solution. Deliverables include the report, code, and presentation.*

## Deliverables

|  |  |
| --- | --- |
| WEEK | DELIVERABLE |
| Week 7 | Proposal Document *- see template* |
| Week 10 | Final Report *- see template*  Presentation slides  Source code |

# **Proposal Document** Template

The proposal should briefly outline **your problem,** **plan**, and **approach**.

Below is the template with examples to guide you:

#### Team Members:

Vann Vat, Chhi HangCheav

#### Project Title:

Delivery System

#### Problem Statement:

Describe the problem you aim to solve and its significance.

Delivery companies face challenges in optimizing their routing to minimize fuel consumption and time. Inefficient routing not only increases expenses but also impacts customer satisfaction and contributes to environmental concerns through higher emissions. Addressing this problem is crucial for businesses aiming to improve operational efficiency and sustainability.

#### Expected Outcomes:

Summarize what you aim to achieve.

The primary goal is to develop a software application that:

1. Recommends optimal delivery routes based on distance and cost.

2. Implements Dijkstra’s algorithm to compute the shortest paths between delivery locations.

3. Evaluates performance using real-world datasets and compares the efficiency of the proposed solution with existing methods.

Expected benefits include reduced delivery times, lower fuel costs, and enhanced overall operational efficiency.

#### Proposed ADT(s):

Identify and justify the choice of ADTs.

Graph-based ADT:

- Nodes: Represent delivery locations.

- Edges: Represent paths between locations, with weights indicating metrics such as distance, fuel cost, or delivery time.

Priority Queue: Used to efficiently determine the next location to process during the execution of Dijkstra's algorithm.

Justification:

- A graph structure is inherently suited for modeling delivery networks, enabling efficient representation of locations and connections.

- Dijkstra's algorithm, a widely used shortest-path algorithm, integrates seamlessly with these ADTs to compute optimal routes efficiently.

#### Implementation Plan:

Outline how you will implement the solution.

Graph Representation:

- Represent the city as a graph using an adjacency list for efficient storage and traversal.

- Nodes correspond to delivery locations, and edges represent paths with associated weights.

Algorithm Selection:

- Implement Dijkstra’s algorithm to calculate the shortest path between a source and destination.

- Use a priority queue to ensure efficient selection of the node with the smallest tentative distance during computation.

Performance Evaluation:

- Test the system with real-world datasets (e.g., city maps, delivery locations).

- Compare the proposed solution with existing methods to validate its efficiency in reducing time and cost.

# **Final Report** Template

The final report should be structured as follows:

#### 1. Introduction

Brief description of the problem and its relevance.

Delivery routing optimization reduces fuel costs and delivery times, directly impacting business efficiency.

#### 2. Problem Definition and Requirements

Define the problem in detail and list any requirements or constraints.

Routes must avoid traffic hotspots and minimize travel time.

#### 3. Abstract Data Type (ADT) Selection

Justify the choice of ADT(s) and how they fit the problem.

Graphs enable efficient representation of routes, and priority queues enhance shortest-path computations.

#### 4. Implementation Details

Provide a thorough explanation of your implementation, covering the following aspects:

##### 4.1 Data Structure(s)

Specify the **data structure(s)** and justify their use in solving the problem.

* Provide a clear API specification for the data structure(s), detailing methods, their parameters, and expected behavior.

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Description | Parameter | Returns |
| shortestPath | Computes the shortest path from a source node to destination node. | Node from  Node to | List of nodes |

* If applicable, include **constraints or limitations** of your data structure(s).
* If needed, include a UML class diagram to **represent relationships** between your classes.

-Graph class with attributes for nodes and edges.

-PriorityQueue class used in conjunction with the Graph for Dijkstra’s algorithm.

##### 4.1 Key algorithms

Describe **1 or 2 key algorithms used** and how they integrate with the data structure(s).

* Input/Output: What the algorithm takes as input and produces as output.
* Step-by-Step Logic: Summarize the main steps and their purpose.

#### 5. Performance Analysis

Analyze the time and space complexity of your solution

Time Complexity: Dijkstra's algorithm runs in O((V + E) log V) for a graph with V vertices and E edges.

Space Complexity: Adjacency list requires O(V + E) space.

#### 6. Results and Discussion

Summarize your findings and compare them with expectations. Include graphs or tables if applicable.

Shortest route computation reduced delivery time by 20%.

#### 7. Challenges and Future Improvements

Discuss challenges faced during the project and potential improvements.

Incorporating real-time traffic data is a challenge we aim to address in future work.

#### 9. References

List any references used during the project.

# **Suggestions of problems to solve**

Here’s a list of suggested problems

You are more than welcome to be creative an come up with your own problem to solve!

Optimal Delivery Routing

Design a system to optimize delivery routes in a city

##### Social Network Analysis:

Find the shortest connection path between two individuals in a social network.

##### Dynamic Traffic Management:

Model a city as a graph and develop a system to reroute traffic dynamically based on congestion data.

##### Spell Checker:

Build a tool that suggests corrections for misspelled words.

##### Task Scheduler:

Develop a system to allocate tasks to workers efficiently, using priority queues or interval scheduling.

##### Conference Room Allocation

Solve the problem of scheduling meetings in the minimum number of rooms using interval graphs.

##### External Sorting:

Implement a sorting algorithm to handle massive datasets that cannot fit in memory.