Pimpri Chinchwad Education Trust's

Pimpri Chinchwad College of Engineering (PCCoE)

(An Autonomous Institute)

Affiliated to Savitribai Phule Pune University(SPPU)

List of assignments – Object Oriented Programming Laboratory

Assignment No. 5 Department : IT Academic year: 2025-26 Sem: 5 Sub: DAA LAB

Aim:

To design and implement an intelligent logistics system that finds the fastest or most cost efficient route for a package in a **multistage** transportation network (warehouses → transit hubs → delivery points).

Objective:

Model the transportation network as a **directed, weighted multistage graph**. • Implement **Dynamic Programming (DP)** or **Dijkstra's Algorithm** to compute the optimal route.

• Ensure the solution **scales** to thousands of cities and routes.

• Adapt to **real-time constraints** like traffic, weather, or fuel efficiency. • Support **batch processing** for multiple delivery requests.

Problem Statement:

SwiftCargo delivers packages across multiple cities using predefined stages: **Source Warehouse → Regional Hub(s) → Transit Hub(s) → Final Delivery Point**. Each package must pass through **at least one node per stage**. There may be multiple alternative routes with different **times/costs** between stages.

The goal is to compute the **optimal route** (minimum time or minimum cost/fuel) from source to destination while:

1. handling **large datasets**, and
2. adapting to **real-time changes** such as traffic jams, road closures, or bad weather.

Outcomes:

• Apply **graph theory** to a real logistics routing problem.

• Implement **DP for multistage graphs** or **Dijkstra's SSSP** for shortest paths. • Design a

solution that **scales** to large networks.

• Integrate **real-time updates** into routing.

- Provide **batch routing** for multiple packages.

Theory:

1. **Graph Representation of a Logistics Network – A Digital Model** Think of the logistics network as a digital map that a computer can understand. • **Nodes (Vertices):** Warehouses, transit hubs, and delivery points.
   - **Edges:** The possible delivery routes connecting two nodes.
   - **Weights:** The cost, travel time, or fuel used to move between two nodes. • **Stages:** The network is divided into stages such as Source → Hub → Transit → Destination. Every package must pass through at least one node in each stage.

Example: A package may start at a Warehouse in Pune (Stage 1) → move to a Hub in Mumbai (Stage 2) → then to a Transit Center in Delhi (Stage 3) → and finally reach the Customer in Chandigarh (Stage 4).

2. **Dynamic Programming in Multistage Graphs – Step-by-Step Route Planner** Dynamic Programming (DP) can be applied when the network is strictly divided into stages.
   - Start from the destination stage, where the cost is 0.
   - Move backwards stage by stage.
   - At each node, calculate the minimum cost = (edge weight + cost of the next stage). • Continue until the source node is reached.
   - The best path is then traced by following the decisions that gave the minimum cost. This

ensures that the route chosen is optimal across all stages.

3. **Dijkstra's Algorithm – The General Optimizer**
   When the network is not strictly multistage, or when real-time updates are required, Dijkstra's Algorithm is used.
   - **Initialization:** `Assign distance = ∞ to all nodes, except the source = 0.` • **Priority Queue:** Keeps track of the next closest node.
   - **Relaxation:** Update the path cost for each neighbor if a shorter path is found. • **Finalization:** Once a node is visited, its shortest distance is locked.

This process continues until the destination is reached, guaranteeing the shortest route.

4. **Handling Real-Time Constraints – Adapting to Change**
   The logistics world is dynamic. Delays or issues can occur suddenly.
   - **Traffic Jam:** Travel times increase for certain routes.
   - **Weather:** Routes may get blocked or slowed down.
   - **Fuel Efficiency:** Routes may be chosen to save cost instead of time.
   - **Rerouting:** If changes occur, the algorithm re-runs with updated data and gives a new path.

This ensures SwiftCargo is always using the best route at that moment.

Algorithm for Finding the Optimal Route – Putting It All Together:

1. Build the network graph by defining all nodes (warehouses, hubs, delivery points) and edges (routes) with their travel costs or times.
2. Run the chosen algorithm: Dynamic Programming (for strict multistage networks) or Dijkstra's

Algorithm (for general networks with real-time changes).

3. After execution, the system finds the minimum cost or travel time for all paths. 4. Trace the path to provide the final delivery route.
5. Keep monitoring traffic or weather updates. If conditions change, re-run the algorithm and update the delivery path.

Questions:

Q1. How do traffic or weather updates affect the delivery routes in SwiftCargo?

Ans: Traffic and weather updates change the road conditions in real time.

These updates affect the edge weights in the route graph:

- ○ Heavy traffic → Increases travel time.
- ○ Bad weather (rain, snow) → Slows down speed or closes roads.

- SwiftCargo's system updates the graph dynamically with new weights from sensors or GPS data.
- The route planner recalculates the best path using algorithms like Dijkstra's, considering the latest conditions.

Q2. Why must each package go through every stage in the SwiftCargo network?

Ans:

- Sorting & Routing – Packages are sorted and sent to the correct destination hub.
- Efficiency – Centralized processing avoids confusion and reduces mistakes.
- Safety & Reliability – Each stage checks handling, prevents loss, and ensures timely delivery.

Conclusion: SwiftCargo's logistics system ensures reliability by adapting routes to external factors like traffic and weather while maintaining a structured multi-stage process for all packages. This combination of flexibility and systematic tracking ensures efficiency, security, and high-quality customer service in large-scale delivery operations.