# Project Description

This project aims to enhance the efficiency and reliability of a transportation network by applying graph theory. By analyzing the network of cities and the routes between them, we will extract valuable insights into route optimization, capacity planning, and cost reduction strategies. A directed, weighted graph representing the transportation routes will be constructed, with various graph algorithms employed to identify critical areas for improvement.

# Graph Construction

* **Vertices (Nodes):** Represent cities as nodes within the network graph.

* **Edges (Links):** Directed edges will connect cities, with weights assigned based on distance.

# Implementation of graph and other algorithms

## Dijkstra's Algorithm (Single Source Shortest Path)

* **Use:** To find the shortest path in terms of time or distance from a city to all other cities.
* **Application:** It will optimize routing for logistics and delivery services, ensuring minimum travel time or distance.

## Floyd-Warshall Algorithm (All Sources Shortest Path)

* **Use:** To compute the shortest paths between all pairs of cities.
* **Application:** Useful for planning multi-stop journeys and understanding the network's efficiency on a larger scale.

## Kruskal's Algorithms (Minimum Spanning Tree, MST)

* **Use:** To connect all cities with the least total distance or time without creating cycles.
* **Application:** Essential for network planning and laying out new routes or services with the least additional infrastructure.

## Prim's Algorithm (Minimum Spanning Tree, MST)

* **Use**: To grow a minimum spanning tree from a root city, adding the most efficient edges.
* **Application**: Helps in identifying the most strategic locations for new service hubs or stations.

## Bellman-Ford Algorithm

* **Use**: To calculate shortest paths, even with some routes having negative factors (like subsidies, discounts).
* **Application**: This can be used to identify whether introducing incentives or subsidies on certain routes could be beneficial overall.

## Breadth-First Search (BFS)

* **Use**: To determine the breadth of connectivity from a given city to all others.
* **Application**: Useful for assessing the accessibility of cities and the immediate reach of the transportation network.

## Depth-First Search (DFS)

* **Use**: To explore all possible paths from one city to another.
* **Application**: Assists in identifying all potential route combinations and can be particularly useful in scenario analysis and disaster recovery planning.

## Implementation in Java

The actual implementation involves creating a Java application that can:

* Parse and load the dataset into a graph data structure with appropriate weights and attributes.
* Implement and run the graph algorithms to analyze and propose optimizations.
* Utilize libraries such as JGraphT for handling graph data and algorithms, and JavaFX or Swing for GUI-based visualization (optional).