Assignment no. 9

Title: Applications of Tree and Graph

- Part 1
    1. Problem Statement: You are tasked with designing a Logistics Tree application for a company that manages the transportation and distribution of goods. The company operates in multiple regions, each with its own set of warehouses and delivery routes.
    2. Requirements:
        a. Region Representation: Implement a tree data structure to represent the hierarchy of regions. Each node represents a region, and child nodes represent sub-regions or warehouses.
        b. Warehouse Management: Allow the user to add, remove, and update information for warehouses (e.g., name, location, capacity).
        c. Route Planning: Design an algorithm to find the most efficient delivery route from one warehouse to another within a region. Use appropriate data structures (e.g., graph) to represent routes.
        d. Inventory Management: Implement operations to add, remove, and update the inventory of goods at each warehouse.
        e. Goods Transfer: Provide functionality to transfer goods between warehouses, ensuring that the transferred quantity does not exceed the capacity of the receiving warehouse.
        f. Cost Calculation: Calculate the cost of transporting goods between warehouses, considering factors like distance, mode of transport, and quantity of goods.
        g. Route Optimization: Optimize routes for multiple deliveries within a region to minimize overall transportation costs and time.
        h. Reporting and Analytics: Generate reports on inventory levels, transportation costs, and delivery times. Implement basic analytics to identify trends and areas for improvement.
        i. User Interface: Develop a user-friendly interface for interacting with the application, allowing users to perform operations easily.

- Part 2
    1. Problem Statement: You are tasked with designing a Train Journey Alternative Route application that helps travelers find alternative routes in case of disruptions or delays in train services.
    2. Requirements:
        a. Graph Representation: Implement a graph data structure to represent the train network. Each node represents a train station, and edges represent the train routes connecting the stations.
        b. Station Information: Allow the user to add, remove, and update information for train stations (e.g., name, location, platform details) as nodes in the graph.
        c. Route Planning: Design an algorithm to find alternative routes between two stations if there are disruptions or delays in the original route. Use graph traversal algorithms like Dijkstra's to find the shortest and most efficient alternative routes.
        d. Disruption Handling: Simulate disruptions or delays in the train network. When a disruption occurs, the application should quickly identify alternative routes and provide them to the user.
        e. Cost Calculation: Calculate the cost of taking alternative routes, considering factors like travel time, number of transfers, and ticket prices.
        f. Route Display: Display the alternative routes to the user, including information such as stations, transfer points, and estimated travel time.
        g. User Interface: Develop a user-friendly interface for travelers to interact with the application, allowing them to input their departure and destination stations, view alternative routes, and get relevant information.