MODULE 5 (B17)

Implementation of Bellman ford algorithm in JAVA

[ASSIGNED TO : N.DAKSHINYA, CB.EN.U4AIE22169]

BELLMAN FORD SHORTEST PATH

Description:

Used to find the shortest paths between vertices in a weighted directed graph. It can handle graphs with negative edge weights and can detect negative cycles in the graph. The Bellman-Ford algorithm has a time complexity of O(V \* E), where V is the number of vertices and E is the number of edges in the graph. It is less efficient than Dijkstra's algorithm for graphs without negative edges, but it can handle negative weights and detect negative cycles.

BellmanFordAlgorithm class:

Represents the implementation of the Bellman-Ford algorithm.

numNodes represents the total number of nodes/landmarks in the graph.

edges is a list that stores the edges (source, destination, weight) of the graph.

The addEdge method adds an edge to the graph by specifying the source, destination, and weight of the edge.

The findShortestPath method takes the start and end nodes as input and calculates the shortest path between them using the Bellman-Ford algorithm.

It returns a list of integers representing the landmarks in the shortest path.

Edge class:

A nested class within BellmanFordAlgorithm that represents an edge in the graph.

Each edge has a source vertex, destination vertex, and weight.

InstructionsForUser class:

Provides a welcome message and displays a table of landmarks in Amrita Vishwa Vidyapeetham.

The LocationTable method creates a table using a two-dimensional array, where each row represents a landmark's number and name.

GetLandmarkName class:

Contains a mapping of landmark numbers to their corresponding names.

The getLandmarkName method takes a landmark number as input and returns the corresponding landmark name.

Motivation :

Implementing a route map navigation system for Amrita Vishwa Vidyapeetham in Coimbatore. The code uses the Bellman-Ford algorithm to find the shortest path between landmarks in the campus.

Can input the landmarks they want to navigate between, and the code will provide the shortest path, ensuring efficient and optimal routing.

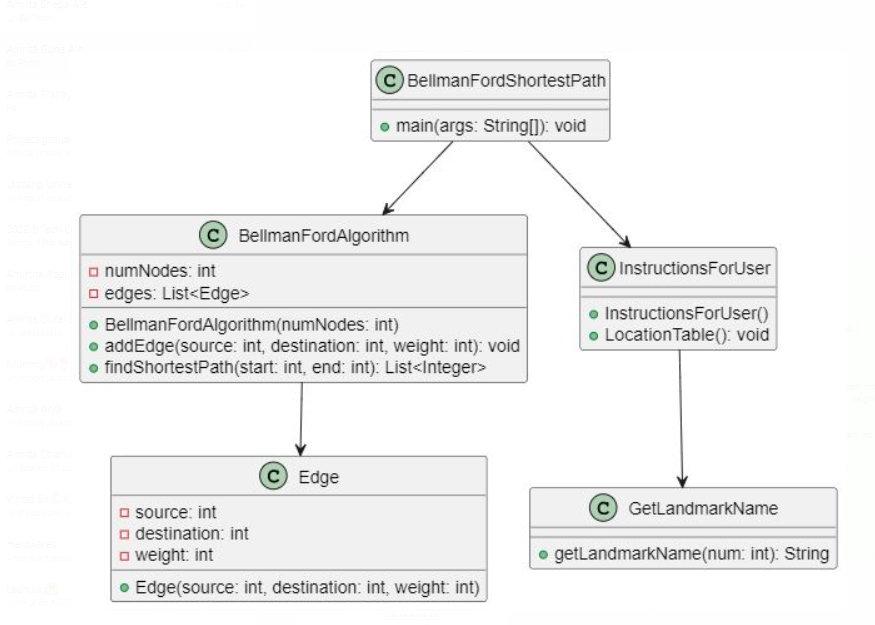
Relevance:

To facilitate navigation and provide directions within the campus by leveraging graph-based algorithms. Users implementation of the Bellman-Ford algorithm for finding the shortest path in a graph. The code specifically focuses on providing a route map navigation system for Amrita Vishwa Vidyapeetham in Coimbatore.

It offers a practical application of the Bellman-Ford algorithm in the context of navigating a campus. By inputting start and end landmarks, users can obtain the shortest path between them, enabling efficient and optimal navigation within the Amrita Vishwa Vidyapeetham campus.The relevance of the code extends to any scenario where route mapping or navigation systems are needed and where the Bellman-Ford algorithm is a suitable solution.

# Input and Output:

UML DIAGRAM



PRESENTED BY:

DAKSHINYA.N

CB.EN.U4AIE22169