**Overview**

This C++ program defines a Graph class and implements Dijkstra's algorithm to find the shortest path from a source vertex to all other vertices in a weighted undirected graph.

**Implementation Details**

**Libraries Used**

* <bits/stdc++.h>: This header includes most of the standard C++ libraries.
* <iostream>: For input and output operations.
* <vector>: To store the distances from the source to each vertex.
* <list>: To store the adjacency list of the graph.
* <queue>: To implement the priority queue used in Dijkstra's algorithm.

**Graph Class**

The Graph class represents a graph using an adjacency list and provides methods to add edges and find the shortest path using Dijkstra's algorithm.

**Class Members**

* int V: The number of vertices in the graph.
* list<myPair> \*adj: Pointer to an array containing adjacency lists.

**Public Methods**

* Graph(int V): Constructor to initialize the graph with V vertices.
* void addEdge(int u, int v, int w): Method to add an edge between vertex u and vertex v with weight w.
* void shortestPath(int src): Method to find the shortest path from the source vertex src to all other vertices using Dijkstra's algorithm.

**Dijkstra's Algorithm**

The shortestPath(int src) method implements Dijkstra's algorithm using a priority queue (min-heap) to efficiently find the shortest path.