Experiment No.6

Title:Shortest Path to other vertices using Dijkstra's algorithm from a given vertex in weighted connected graph.

Problem:- Urban Traffic Management

Optimizing traffic flow by determining shortest paths for vehicles under varying traffic weights. cive code and its output for this problem

Programm:-
```
#include <stdio.h>

#include <limits.h>

#define V 6 // Number of intersections (vertices)

// Function to find the vertex with minimum distance

int minDistance(int dist[], int visited[]) {

    int min = INT_MAX, min_index = -1;

    for (int v = 0; v < V; v++) {

        if (!visited[v] && dist[v] <= min) {

            min = dist[v];

            min_index = v;

        }

    }

    return min_index;

}

// Function to print distances

void printSolution(int dist[]) {

    printf("Intersection\tShortest Traffic Time from Source\n");

    for (int i = 0; i < V; i++)

        printf("%d\t\t%d\n", i, dist[i]);
```

```
}
// Dijkstra's algorithm
void dijkstra(int graph[V][V], int src) {
    int dist[V];    // Shortest distances from source
    int visited[V]; // Visited intersections
    for (int i = 0; i < V; i++) {
        dist[i] = INT_MAX;
        visited[i] = 0;
    }
    dist[src] = 0;
    for (int count = 0; count < V - 1; count++) {
        int u = minDistance(dist, visited);
        visited[u] = 1;
        for (int v = 0; v < V; v++) {
            if (!visited[v] && graph[u][v] && dist[u] != INT_MAX
                && dist[u] + graph[u][v] < dist[v]) {
                dist[v] = dist[u] + graph[u][v];
            }
        }
    }
    printSolution(dist);
}
int main() {
    // Traffic network graph (weights = traffic time in minutes)
    int graph[V][V] = {
        {0, 4, 0, 0, 0, 0},
```

```
    {4, 0, 8, 0, 0, 0},
    {0, 8, 0, 7, 0, 4},
    {0, 0, 7, 0, 9, 14},
    {0, 0, 0, 9, 0, 10},
    {0, 0, 4, 14, 10, 0}
  };
  int source = 0; // Starting intersection
  printf("Urban Traffic Management - Shortest Paths from Intersection %d\n", source);
  dijkstra(graph, source);
  return 0;
}
```

Output:-

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS                Code  +

PS C:\Users\shiva> cd "c:\c language\" ; if ($?) { gcc DAAappl_1.c -o DAAappl_1 } ; if ($?) { .\DAAappl_1 }
Urban Traffic Management - Shortest Paths from Intersection 0
Intersection    Shortest Traffic Time from Source
0               0
1               4
2               12
3               19
4               26
5               16
PS C:\c language>
```

Real-World Applications of Dijkstra's Algorithm

1. GPS Navigation and Route Planning
   - Finding the shortest or fastest route from your location to multiple destinations.
   - Used in Google Maps, Waze, and car navigation systems.
2. Telecommunication Networks
   - Determining the least-cost routing of data packets in networks like the Internet.
   - Optimizes latency and bandwidth usage in network routing protocols (e.g., OSPF).
3. Transportation and Logistics

- Calculating minimum travel cost or time for trucks, delivery vans, or public transport.
- Optimizes routes for supply chains and delivery networks.

4. Airline Route Optimization
    - Finding the shortest or least expensive path connecting multiple airports.
    - Useful for minimizing travel time, fuel cost, or layovers.
5. Robot Path Planning
    - Enabling robots or drones to move efficiently through weighted grids or maps, avoiding obstacles.
6. Urban Traffic Management
    - Optimizing traffic flow by determining shortest paths for vehicles under varying traffic weights.
7. Computer Networks
    - Shortest path for message passing in distributed systems.
    - Efficient routing in ad-hoc wireless or sensor networks.
8. Game Development
    - Calculating AI character movement on weighted maps for shortest and optimal paths.