ASSIGNMENT 5

# AIM:

Consider the scenario of the supply chain management system. The supply chain involves multiple locations, such as suppliers, warehouses, distribution centers, and retail stores, connected by transportation networks. Find the shortest possible route using Dijkstra's algorithm to optimize transportation path and inventory management.

# SOURCE CODE:

#include <iostream> #include <vector> #include <queue> using namespace std;

const int INF = 1e9;

vector<pair<int, int>> adj[100]; // Adjacency list (node -> {neighbor, weight})

void dijkstra(int src, int n) {

vector<int> dist(n, INF); // Distance vector

priority\_queue<pair<int, int>, vector<pair<int, int>>, greater<pair<int, int>>> pq;

dist[src] = 0; pq.push({0, src});

while (!pq.empty()) {

int d = pq.top().first; int u = pq.top().second; pq.pop();

if (d > dist[u]) continue; // Outdated entry, skip for (auto edge : adj[u]) {

int v = edge.first;

int w = edge.second;

if (dist[u] + w < dist[v]) { dist[v] = dist[u] + w; pq.push({dist[v], v});

}

}

}

// Print distances

cout << "Shortest distances from node " << src << ":\n"; for (int i = 0; i < n; i++) {

if (dist[i] == INF)

cout << "Node " << i << " is unreachable\n"; else

cout << "Distance to node " << i << " : " << dist[i] << endl;

}

}

int main() {

int n = 3; // Total number of nodes

// Directed edges: adj[u].push\_back({v, weight}); adj[0].push\_back({1, 4});

adj[0].push\_back({2, 1});

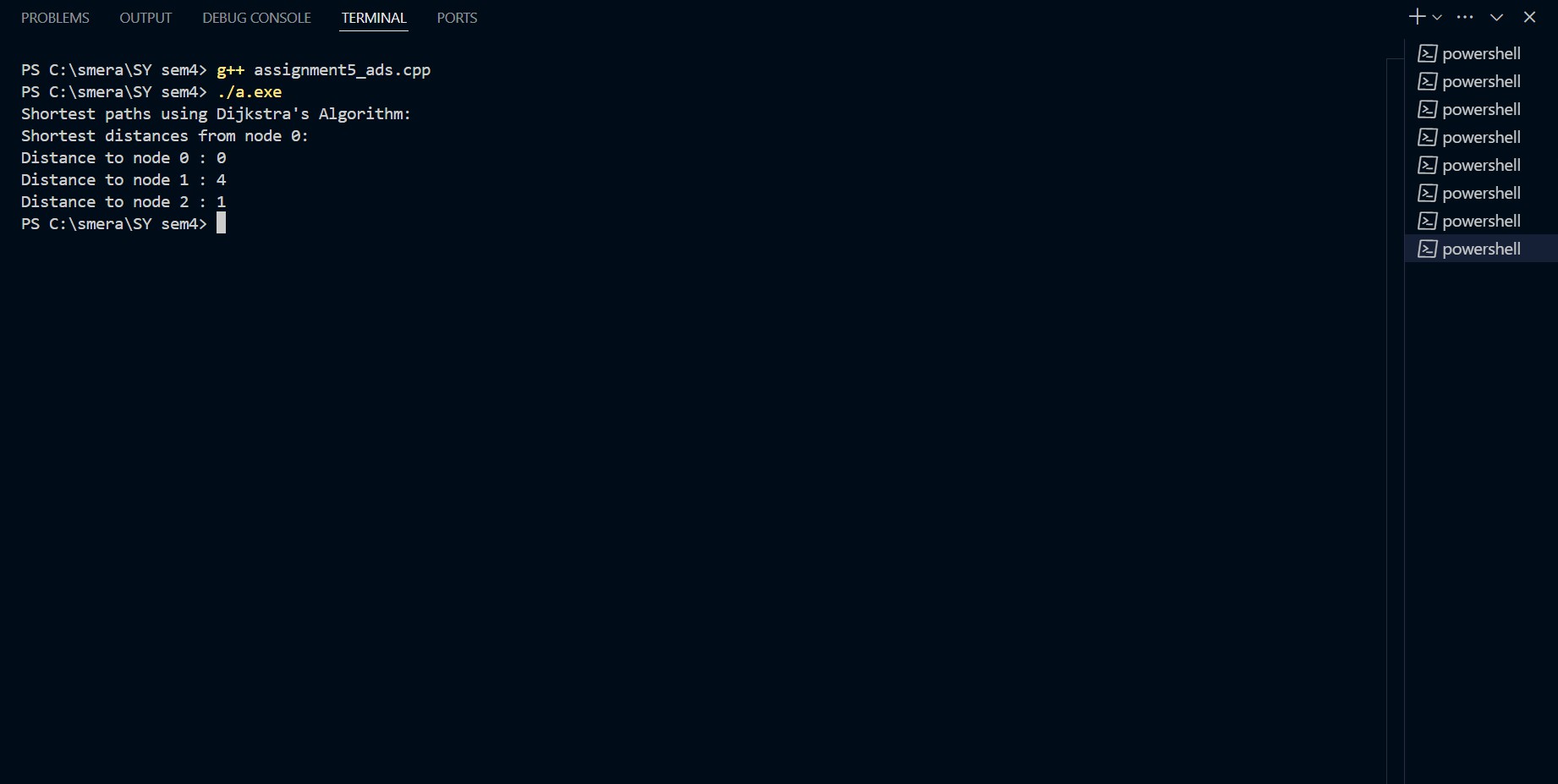
adj[1].push\_back({2, 2});

cout << "Shortest paths using Dijkstra's Algorithm:\n"; dijkstra(0, n); // Starting from node 0

return 0;

}

# OUTPUT:

****

**CONCLUSION:**

The implementation of Dijkstra’s algorithm in supply chain management helps optimize transportation routes, ensuring cost-effective and timely deliveries. By representing locations as graph nodes and paths as edges, the algorithm efficiently finds the shortest route between different supply points. This improves logistics, minimizes delays, and enhances overall supply chain efficiency, making it a valuable approach for real-world applications.