231501058 CS23231 – D a t a S t r u c t u r e s

**Ex. No.: 14**

**Graph Traversal**

**Date: 31/5/24**

**Write a C prog ram to create a g raph and find the shortest path using Dijikstra’s Alg orithm .**

**Algorithm:**

#include <stdio.h>

#include <limits.h>

#define MAX\_VERTICES 100

int minDistance(int dist[], int sptSet[], int vertices) { int min = INT\_MAX, minIndex;

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

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

min = dist[v];

minIndex = v;

}

}

return minIndex;

}

void printSolution(int dist[], int vertices) { printf("Vertex \tDistance from Source\n"); for (int i = 0; i < vertices; i++) {

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

}

}

void dijkstra(int graph[MAX\_VERTICES][MAX\_VERTICES], int src, int vertices) { int dist[MAX\_VERTICES];

int sptSet[MAX\_VERTICES];

for (int i = 0; i < vertices; i++) {

dist[i] = INT\_MAX;

sptSet[i] = 0;

}

dist[src] = 0;

for (int count = 0; count < vertices - 1; count++) {

int u = minDistance(dist, sptSet, vertices); sptSet[u] = 1;

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

if (!sptSet[v] && graph[u][v] && dist[u] != INT\_MAX && dist[u] + graph[u][v] < dist[v]) {

dist[v] = dist[u] + graph[u][v];



**Dept of Artificial Intelligence and Machine Learning** | **Rajalakshmi Engineering College**

. 49

231501058 CS23231 – D a t a S t r u c t u r e s

}

}

}

printSolution(dist, vertices);

}

int main() {

int vertices = 5;

int graph[MAX\_VERTICES][MAX\_VERTICES] = {

{0, 10, INT\_MAX, 30, 100},

{10, 0, 50, INT\_MAX, INT\_MAX},

{INT\_MAX, 50, 0, 20, 10},

{30, INT\_MAX, 20, 0, 60},

{100, INT\_MAX, 10, 60, 0}

};

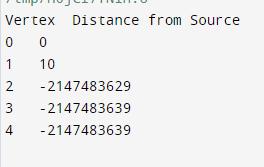
int source = 0;

dijkstra(graph, source, vertices);

return 0;

}

**OUTPUT**



**Dept of Artificial Intelligence and Machine Learning** | **Rajalakshmi Engineering College**

. 50