**DIJKSTRAS**

#include <stdio.h>

#include <limits.h>

#define MAX\_VERTICES 10

#define INF INT\_MAX

int minDistance(int dist[], int sptSet[], int V) {

int min = INF, min\_index;

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

if (sptSet[v] == 0 && dist[v] <= min) {

min = dist[v];

min\_index = v;

}

}

return min\_index;

}

void dijkstra(int graph[MAX\_VERTICES][MAX\_VERTICES], int V, int src) {

int dist[MAX\_VERTICES];

int sptSet[MAX\_VERTICES];

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

dist[i] = INF;

sptSet[i] = 0;

}

dist[src] = 0;

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

int u = minDistance(dist, sptSet, V);

sptSet[u] = 1;

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

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

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

}

}

}

printf("Vertex \tDistance from Source %d\n", src);

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

if (dist[i] == INF) {

printf("%d \tINF\n", i);

} else {

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

}

}

}

int main() {

int V, E;

printf("Enter the number of vertices: ");

scanf("%d", &V);

int graph[MAX\_VERTICES][MAX\_VERTICES] = {0};

printf("Enter the number of edges: ");

scanf("%d", &E);

printf("Enter the edges (u, v, weight) for each edge (0-indexed):\n");

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

int u, v, weight;

scanf("%d %d %d", &u, &v, &weight);

graph[u][v] = weight;

graph[v][u] = weight; // For undirected graph

}

int src;

printf("Enter the source vertex: ");

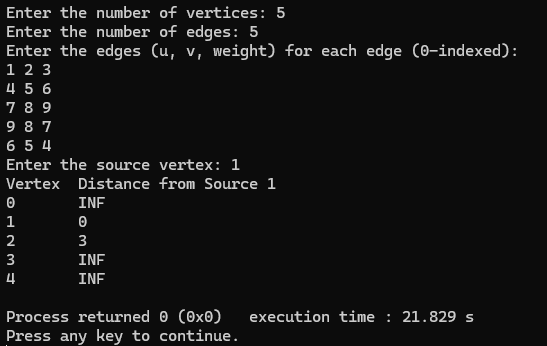
scanf("%d", &src);

dijkstra(graph, V, src);

return 0;

}

**OUTPUT:**

****