**PROGRAM 16: Dijkstra’s algorithm**

From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra’s algorithm.

**Procedure:**

1) Create a set sptSet (shortest path tree set) that keeps track of vertices included in shortest path tree, i.e., whose minimum distance from source is calculated and finalized. Initially, this set is empty.

2) Assign a distance value to all vertices in the input graph. Initialize all distance values as INFINITE. Assign distance value as 0 for the source vertex so that it is picked first.

3) While sptSet doesn’t include all vertices

….a) Pick a vertex u which is not there in sptSet and has minimum distance value.

….b) Include u to sptSet.

….c) Update distance value of all adjacent vertices of u. To update the distance values, iterate through all adjacent vertices. For every adjacent vertex v, if sum of distance value of u (from source) and weight of edge u-v, is less than the distance value of v, then update the distance value of v.

**Program:**

#include <stdio.h>

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

{

int min = 999, min\_index;

int v;

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

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

min = dist[v], min\_index = v;

return min\_index;

}

int printSolution(int src,int dist[],int V )

{ int i;

printf("\n\t Vertex \t\t Distance from Source\n\n");

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

printf("\t%c ----> %c \t\t\t %d\n",src+65, i+65, dist[i]);

}

void dijkstra(int graph[10][10], int src,int V )

{

int dist[V];

int i,count,u,v;

int sptSet[V];

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

dist[i] = 999, sptSet[i] = 0;

dist[src] = 0;

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

u = minDistance(dist, sptSet,V);

sptSet[u] = 1;

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

if (!sptSet[v] && graph[u][v] && dist[u] != 999

&& dist[u] + graph[u][v] < dist[v])

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

}

printSolution(src,dist,V);

}

int main()

{

int i,j,V;

int graph[10][10];

printf("Enter number of vertices\n");

scanf("%d",&V);

printf("Enter adjacency matrix\n");

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

{

for(j=0;j<V;j++)

scanf("%d",&graph[i][j]);

}

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

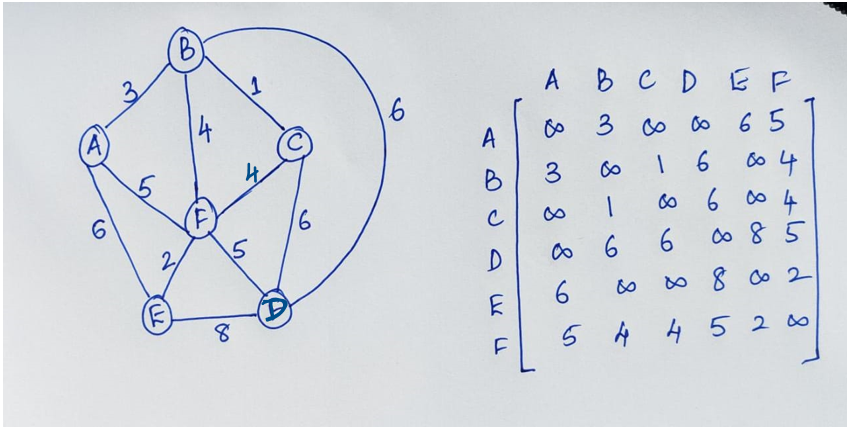
dijkstra(graph,i,V );

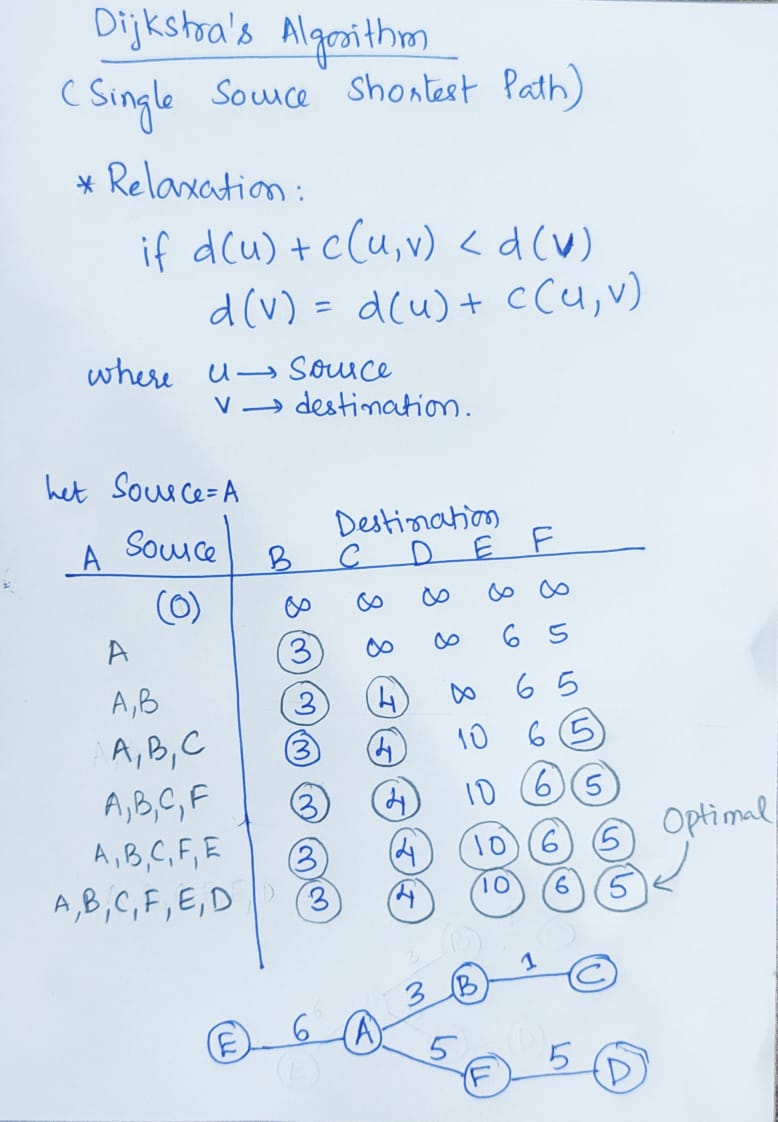
}

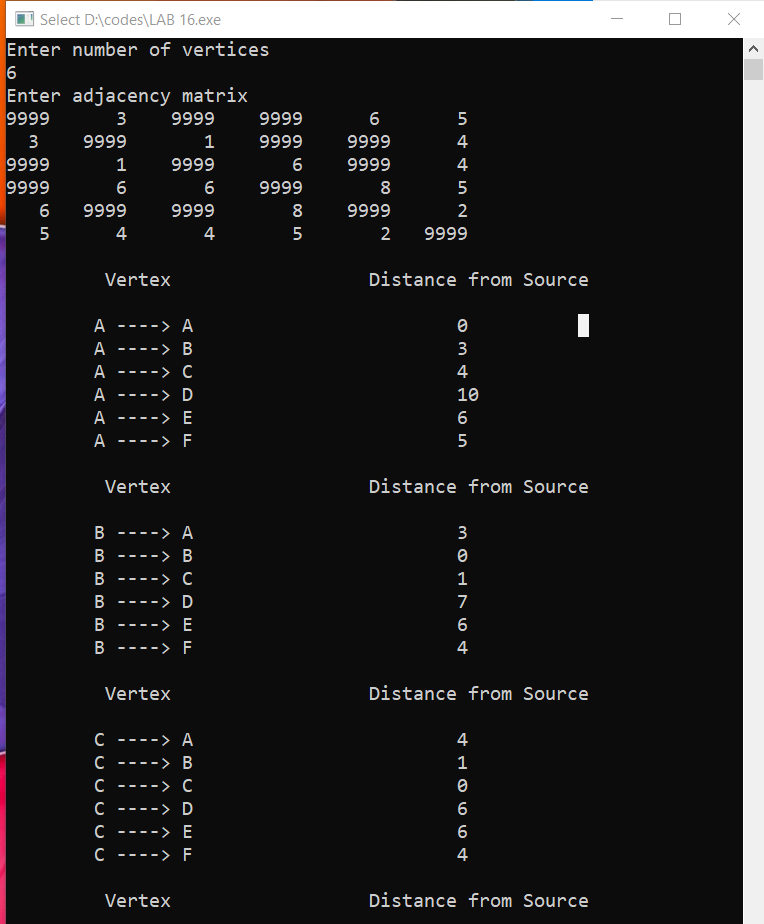
// dijkstra(graph, 0,V );

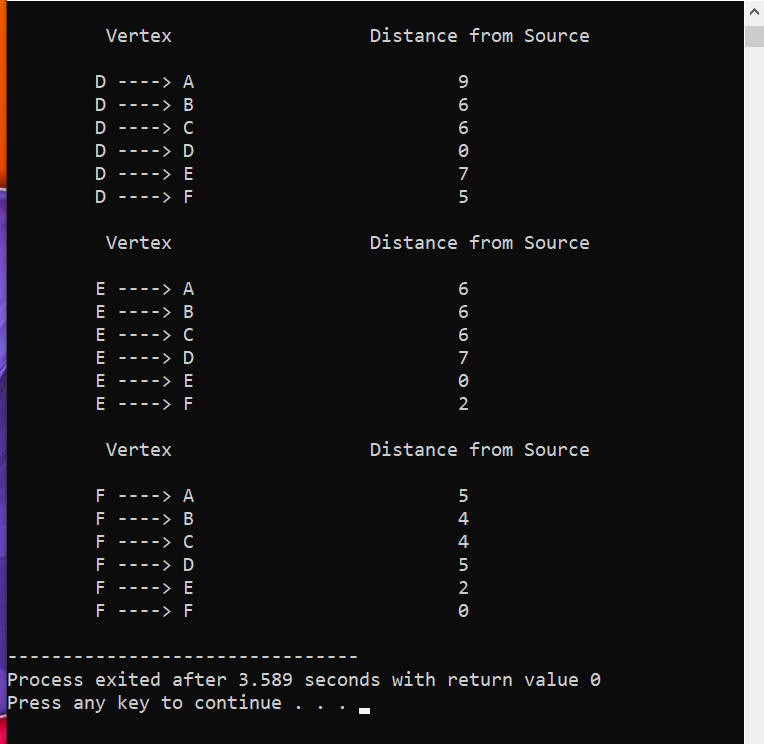
return 0;

}

****



****

****

Reference:

[Dijsktra's algorithm (geeksforgeeks.org)](https://www.geeksforgeeks.org/dijkstras-shortest-path-algorithm-greedy-algo-7/)