**Dijikstra’s Algorithm**

**R. Harini**

**18BCE1010**

**Code:**

#include <limits.h>

#include <stdio.h>

#define V 6

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

{

int min = INT\_MAX, 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;

}

int printSolution(int dist[])

{

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

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

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

}

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

{

int dist[V];

int sptSet[V];

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

dist[i] = INT\_MAX, sptSet[i] = 0;

dist[src] = 0;

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

int u = minDistance(dist, sptSet);

sptSet[u] = 1;

for (int v = 0; v < V; 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];

}

printSolution(dist);

}

int main()

{

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

int a;

scanf("%d",&a);

int graph[V][V];

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

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

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

}

}

dijkstra(graph, 0);

return 0;

}

