DATA STRUCTURES

DAY-14

1. Shortest path algorithm

Program:

```
#include <stdio.h>

#include <limits.h>

#include <stdbool.h>

#define V 5 int minDistance(int dist[], bool sptSet[]) {

  int min = INT_MAX, min_index;

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

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

      min = dist[v], min_index = v;

  return min_index;

}

void printSolution(int dist[]) {

  printf("Vertex \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];

  bool sptSet[V];

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

    dist[i] = INT_MAX, sptSet[i] = false;


  dist[src] = 0;
```

```
    for (int count = 0; count < V - 1; count++) {
        int u = minDistance(dist, sptSet);
        sptSet[u] = true;
        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() {
    int graph[V][V] = {
        {0, 10, 0, 30, 100},
        {10, 0, 50, 0, 0},
        {0, 50, 0, 20, 10},
        {30, 0, 20, 0, 60},
        {100, 0, 10, 60, 0},
    };
    dijkstra(graph, 0);
    return 0;
}
```

Output:

```
Vertex   Distance from Source
```

```
0       0

1       10

2       50

3       30

4       60
```

2.Dijkstra’s Algorithm

Program:

```
#include <stdio.h>
#include <limits.h>
#define V 5
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;
}
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];
            }
        }
    }
    printf("Vertex \t Distance from Source\n");
    for (int i = 0; i < V; i++) {
        printf("%d \t %d\n", i, dist[i]);
    }
}
int main() {
    int graph[V][V] = { {0, 10, 0, 30, 100},
                        {10, 0, 50, 0, 0},
                        {0, 50, 0, 20, 10},
                        {30, 0, 20, 0, 60},
                        {100, 0, 10, 60, 0} };
    dijkstra(graph, 0);
    return 0;
}
```

Output:

```
Vertex   Distance from Source

0        0

1        10

2        50

3        30

4        60
```