Lab 09 Dijkstra’s Algorithm

Task: In the last lab we implemented DFS and BFS using both adjacency matrix & adjacency list. Now we are going to continue where we left of. You are required to implement Dijkstra’s algorithm of finding shortest path. You have to do this via both:

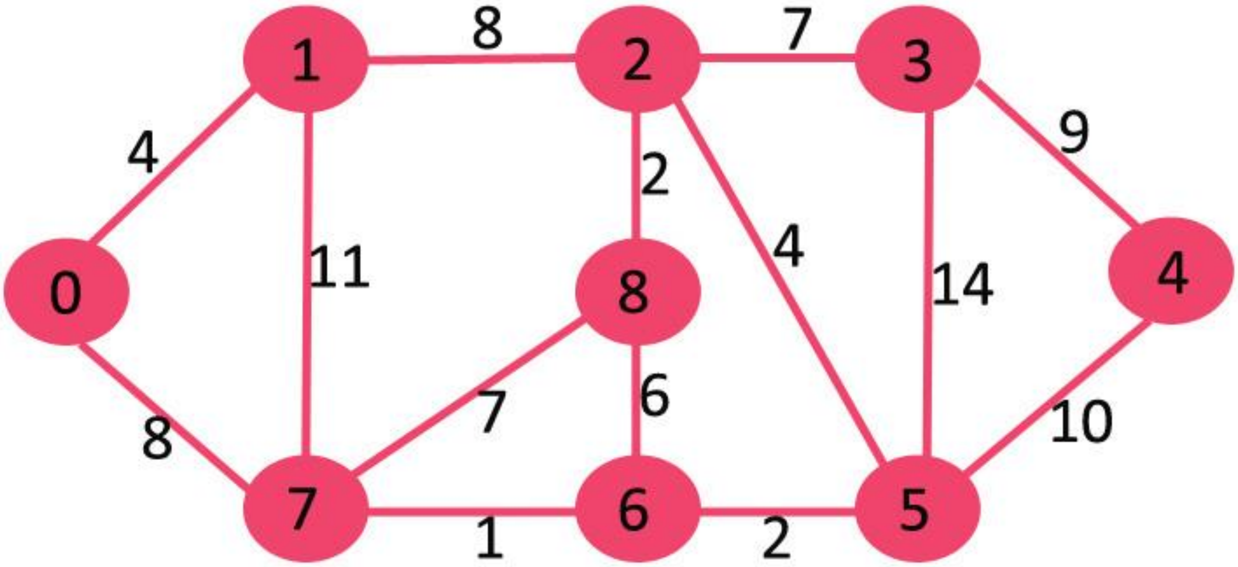
1- Adjacency matrix
2- Adjacency list

Input: The input “Graph [] []” should be given as a 2D array. Whether you take this as hard coded or input from user is your own choice. There should also be only one final/goal node “E”. The 2D array should look like this:

```
int graph[V][V] = {
                        { 0, 4, 0, 0, 0, 0, 0, 8, 0 },
                        { 4, 0, 8, 0, 0, 0, 0, 11, 0 },
                        { 0, 8, 0, 7, 0, 4, 0, 0, 2 },
                        { 0, 0, 7, 0, 9, 14, 0, 0, 0 },
                        { 0, 0, 0, 9, 0, 10, 0, 0, 0 },
                        { 0, 0, 4, 14, 10, 0, 2, 0, 0 },
                        { 0, 0, 0, 0, 0, 2, 0, 1, 6 },
                        { 8, 11, 0, 0, 0, 0, 1, 0, 7 },
                        { 0, 0, 2, 0, 0, 0, 6, 7, 0 }
                   };
```

…where V is the number of vertices/nodes.

By: Tariq Aman



Output: The output on the screen should be the individual distance of each vertex from the goal node "E". It should look like this:

```
Vertex   Distance from Source
0            0
1            4
2            12
3            19
4            21
5            11
6            9
7            8
8            14
```

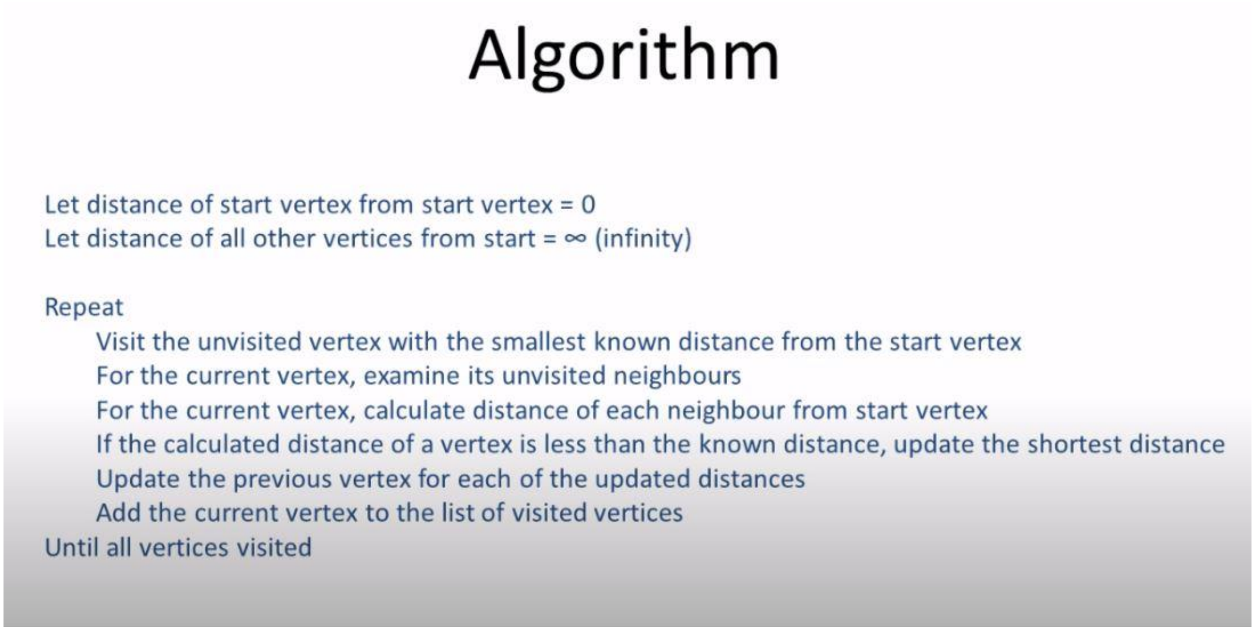
Function: The function "Dijkstra ()" takes in two parameters i.e. "Graph [] []" and "start". Here the start represents the starting/parent node.

void Dijkstra (Graph [] [], start, end)

By: Tariq Aman

Steps for Dijkstra’s Algorithm:

To find shortest path to one desired node in a graph, follow these steps:



Algorithm

```
Let distance of start vertex from start vertex = 0
Let distance of all other vertices from start = ∞ (infinity)

Repeat
    Visit the unvisited vertex with the smallest known distance from the start vertex
    For the current vertex, examine its unvisited neighbours
    For the current vertex, calculate distance of each neighbour from start vertex
    If the calculated distance of a vertex is less than the known distance, update the shortest distance
    Update the previous vertex for each of the updated distances
    Add the current vertex to the list of visited vertices
Until all vertices visited
```