**Department of Computer Science and Engineering**

|  |  |
| --- | --- |
| **Course Code:CSE221** | **Credits: 1.5** |
| **Course Name: Algorithms** | **Semester: Fall’18** |

**Lab 07  
Single Source Shortest Path**

**(Dijkstra’s)**

1. **Topic Overview:**

The idea of Single source shortest path algorithm is to find the shortest path from a source vertex to all/ any other vertex in connected **weighted** graph. The shortest path scenario is something we encounter every day when we set out from home on our way to university and there are multiple routes. If the weights are considered as traffic level the path with the minimum number of total traffic will be extracted using this algorithm.

Dijkstra’s algorithm is a greedy approach to find shortest path in a connected weighted graph. We follow greedy approach so that we can use the heuristics connected to the problem which aid us to come up with a solution without roaming around the whole graph. It reduces the computation for those parts of the graph which has less possibilities of being a part of the solution to the problem defined.

1. **Lesson Fit:**

To solve this problem, the students must have a basic idea on the following concepts:

* 1. Graph Representation and Computation
  2. Greedy Algorithm
  3. Optimization Task

1. **Learning Outcome:**

After this lab, the students will be able to:

* 1. Find shortest path from a connected graph from a given source
  2. Represent real life problem in graphs
  3. Optimize the calculation for finding solution

1. **Anticipated Challenges and Possible Solutions**
2. **Acceptance and Evaluation**

Students will show their progress as they complete each problem. They will be marked according to their class performance. There may be students who might not be able to finish all the tasks, they will submit them later and give a viva to get their performance mark. The mark distribution for the lab will be as follows:

Code: 05

Viva: 05

1. **Activity Detail**
   1. **First 1.5 hours  
      Explanation:**The lab instructor will explain how to represent a connected graph in a matrix and traversing the graph using the matrix. Instructor will explain the Dijkstra’s algorithm in a connected graph and will let the students understand the approach thoroughly.
   2. **Second 1.5 hours**

**Implementation:**

After explanation, the students will implement Prim’s algorithm to solve a given problem.

**Problem Task: See Activity**

1. **Home tasks:** Unfinished tasks

**Lab 7 Activity**

**Problem Description:** A portion of the map of Dhaka is given in the picture.

There are 2 mother nodes Motijheel, which is the source, and Moghbazar the destination. The other nodes from A to L represent intersections. There are multiple routes to reach from source to destination. The table below shows the weights of each route which represent the level of traffic. The higher the value, higher the traffic. Using your knowledge on graph, implement Dijkstra’s algorithm. Print the output, i.e. the level of traffic to reach Moghbazar from Motijheel.

**Note:** BFS also gives the shortest path between source and destination. Why not use BFS in this situation?

**Input Format:**

First line will contain source destination

Second line will contain “n” strings/Character which represents the name of each intersection separated by space

Next “m” lines will contain the edges and the level of traffic in the following format “node1 node2 traffic” (e.g. Motijheel A 3)

|  |  |  |
| --- | --- | --- |
| **Vertex 1** | **Vertex 2** | **Traffic Level** |
| Motijheel | A | 3 |
| A | B | 4 |
| A | H | 6 |
| B | G | 2 |
| H | I | 5 |
| B | C | 7 |
| C | F | 7 |
| C | D | 3 |
| D | E | 1 |
| F | G | 2 |
| F | MOGHBAZAR | 4 |
| G | H | 3 |
| G | J | 1 |
| I | J | 7 |
| J | K | 6 |
| K | L | 4 |
| K | MOGHBAZAR | 7 |
| L | MOGHBAZAR | 2 |

**Output:**

Output the **shortest path** and **cost of path** i.e. minimum traffic from source to destination (e.g source🡪A🡪Z🡪Y🡪 destination: X)



