***Name**: Souham Biswas* ***CWID**: A20365242* ***Email**: sbiswas7@hawk.iit.edu*

CS 542 – Computer Networks I: Fundamentals

Project Operations Manual

Program Design-

The programming language used was C++ in a Windows environment. The code is organized in specific classes performing specific functions. The following classes are provided –

- **FileProcessor-**
  This class encapsulates functionality to process input text files containing details of the adjacency matrix which are parsed to generate the network graphs for link state router simulation. This class contains the following methods and properties –

  Public Methods -

  - **`LoadFile:`** This method takes the filename string as a constant pointer to character array. It reads the file line by line and stores the read lines as a string vector which are later converted to integers for a consistent memory representation usable for simulation.
  - **`GetLinkMatrix:`** Allocates memory to store adjacency matrix and populates it with values read from the input text file.
  - **`PrintLinkMatrix:`** Prints the 2D adjacency matrix to standard out.

  Public Variables –

  - **`num_routers:`** Number of routers in the adjacency matrix; that is, edge length of the adjacency matrix.
  - **`link_matrix:`** Double pointer to the memory representation of the 2D adjacency matrix.
  - **`file_input:`** Vector of strings where each element is a string representation of each row of the adjacency matrix as read from it.

  Private Methods –

  - **`split_:`** This method has two overloaded implementations, one is a worker and the other is the entry point. This method splits the input string into a vector of strings based upon the input character delimiter. It is used to read the text from the input file into the memory representation of the adjacency matrix.

  Private Variables –

  - **`network_def_file_stream:`** File stream object used to read input text file with the adjacency matrix.

- **Router-**
  This class encapsulates the functionality of a single router object in a graph of Router objects which is contained in a Network Object. Each Router object basically contains information about it's own ID, a list of references to adjacent Router objects in the Graph along with the shortest paths and their costs to every other Router object. It also contains the broadcast cost of the respective Router object.

  Example usage: `Router *router = new Network(3, 4);`

  This example creates a Router object with Router ID 3 (1 indexed) to be used in a Network Object having a graph of 4 Router Objects.

  Public Methods -

  - **`Router:`** This is the class constructor which simply updates object properties with the router ID & the number of routers in the network of which the given router is a part.

  Public Variables –

  - **`router_id:`** Number of routers in the adjacency matrix; that is, edge length of the adjacency matrix.
  - **`num_routers_in_network:`** Total number of routers in the network of which this router is a part..
  - **`broadcast_cost:`** Sum of shortest path costs from this router to all other routers in the Network graph of Router objects.
  - **`connections:`** Sum of shortest path costs from this router to all other routers in the Network graph of Router objects.
  - **`in_costs:`** Costs of links incoming to the Router object from adjacent Router objects in the Network graph.
  - **`out_costs:`** Costs of links outgoing to the adjacent Router objects in the Network graph.
  - **`all_path_costs:`** Vector of shortest path costs to all other Router objects in the Network graph.
  - **`shortest_paths:`** Vector of Lists where each list of Router IDs represents the shortest path to the target Router. The target Router's ID is the Index of the list in the Vector plus 1 (as Router IDs are 1 indexed). Each list is organized in a source ---> destination fashion; that is, the start() iterator will return the source Router ID (which is this one) and the end() iterator returns the destination Router ID (where the path terminates)
  - **`connection_table:`** Connection table of the given Router. This maps every router in the Network to the Router ID of one of the adjacent Routers of this Router in the Network graph which every packet must be immediately redirected to ensure a minimum transmission cost. The keys are target Router ID – 1 and the values are the Router IDs of the immediately next Router where the packet is to redirected.

- **Network-**
  This class encapsulates the network architecture of the Routers for simulating Link State Routing Protocol. A Network object actually creates a graph of Router Objects. The CreateNetwork method is used to instantiate a Network object which accepts an adjacency matrix (as a 2D matrix) and the number of routers as arguments.

  Example usage:
  ```
  Network *network = new Network();
  network->CreateNetwork(link_matrix, num_routers);
  ```

  Public Methods -

  - **`CreateNetwork:`** Method to create Network Graph from adjacency matrix as read from input file.
  - **`PopulateRouterConnectionTable:`** Populates the connection table of the router pertaining to the router ID specified in argument. Other things also populated include the list of shortest paths and their costs to every other router in the network
  - **`RemoveRouter:`** Simulates a router which is malfunctioning in the network. This method in effect, modifies the original adjacency matrix by replacing all the connections to the input router with negative weight edges which symbolize no edges and rebuilds the graph of router objects.
  - **`FindBroadcastRouter:`** Updates the `broadcast_router_id` property of the network object. This method finds the router most suitable for a broadcast operation in the network; it basically finds the router in the network which has a minimum sum of costs of paths to every router in the network

  Public Variables –

  - **`link_matrix:`** Double pointer to the adjacency matrix describing the network topology represented in memory as a 2D matrix (Array of pointers to arrays).
  - **`num_routers:`** Number of routers in the network.
  - **`broadcast_router_id:`** Router ID of broadcast router. Broadcast router is the router in the network which has the minimum sum of shortest paths to all other routers in the network; hence, it's most suitable for a broadcast operation.
  - **`routers:`** Vector of actual Router objects which are used to create a graph of Router objects simulating a network.
  - **`router_down:`** Boolean array used to keep track of defective routers which cannot be used in the network. All values initially set to false.

  Private Methods –

  - **`DijkstrasAlgorithm:`** Contains implementation of the Dijkstra's algorithm to find shortest paths to all nodes from a given source node in a graph. It takes the Router ID of the source router as an input and updates the corresponding Router object's following properties - `connection_table`, `shortest_paths`, `all_path_costs`
  - **`UpdateCosts:`** Used in the implementation of Dijkstra's algorithm. This method updates the path costs to the routers immediately adjacent to the input router.

Private Variables –

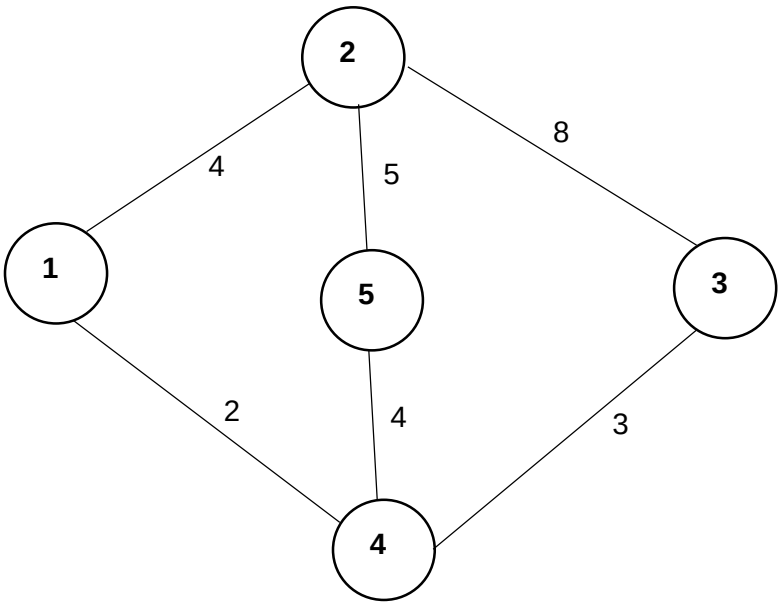
- **`first_init`:** Is true only when the network is initialized for the first time. This flag is used to initialize the `router_down` boolean array keeping track of 'defective' routers..

Program Working Verification-

To test the correctness of the program, the following adjacency matrix was firstly used as an input as it is easy to visualize –

```
0       4       -1      2       -1
4       0       8       -1      5
-1      8       0       3       -1
2       -1      3       0       4
-1      5       -1      4       0
```

The given adjacency matrix can be visualized in the following way –



The edge costs are represented by the numbers on the edges and the router IDs are represented by the numbers inside the nodes. The results obtained from the program are given below -

```
Router 1 Connection Table
Destination     Interface
========================
1               -
2               2
3               4
4               4
5               4
```

```
Router 2 Connection Table
Destination     Interface
========================
1               1
2               -
3               3
4               1
5               5

Router 3 Connection Table
Destination     Interface
========================
1               4
2               2
3               -
4               4
5               4

Router 4 Connection Table
Destination     Interface
========================
1               1
2               1
3               3
4               -
5               5

Router 5 Connection Table
Destination     Interface
========================
1               4
2               2
3               4
4               4
5               -
```

The shortest path obtained from router 1 to router 3 as a result of command 3 is given as –

```
Shortest Path from router 1 to router 3 is -
1 -> 4 -> 3
Cost = 5
```

Now upon removing router 4 from the network using command 4, the new shortest path obtained between routers 1 & 3 is the following –

```
Shortest Path from router 1 to router 3 is -
1 -> 2 -> 3
Cost = 12
```

Notice that the path cost increased which verifies that the least cost path in the network is being computed. The new connection table of router 1 is as follows –

```
Router 1 Connection Table
Destination     Interface
=======================
1               -
2               2
3               2
4               -
5               2
```

Results obtained for command 5 to find the best router for broadcast are presented below –

```
Broadcast Router is Router 4
Broadcast Cost (sum of path costs to all other routers) = 15
Router 4 Connection Table
Destination     Interface
=======================
1               1
2               1
3               3
4               -
5               5
```

Broadcast costs from all the routers are given below –

Broadcast Router ID	Broadcast Cost
1	17
2	23
3	23
4	15
5	22

As is clearly evident, broadcast cost from router 4 is the minimum which is returned. This verifies the working of command 5 to find the best router for broadcast.

Tests with other inputs-

Results obtained for a network with 20 routers is as follows (all commands shown below follow the same sequence as presented here) –

```
-----------------------------------
CS542 Link State Routing Simulator

(1) Create a Network Topology
(2) Build a Connection Table
(3) Shortest Path to Destination Router
(4) Modify a Topology
(5) Best Router for Broadcast
(6) Exit

Command: 1

Input original network topology matrix data file: matrix_20.txt
Review original topology matrix:
0   -1  -1  -1  -1  -1  5   1   -1  -1  -1  -1  -1  2   -1  -1  -1  4   5   4
-1  0   -1  3   -1  -1  -1  -1  -1  -1  -1  -1  -1  5   -1  -1  2   -1  2   2
-1  -1  0   -1  4   -1  2   -1  -1  1   -1  -1  -1  -1  -1  5   -1  -1  -1  -1
-1  3   -1  0   5   -1  1   2   -1  -1  3   -1  -1  -1  -1  -1  -1  -1  3   -1
-1  -1  4   5   0   -1  2   -1  -1  -1  -1  2   -1  -1  -1  -1  1   1   -1  5
-1  -1  -1  -1  -1  0   3   -1  -1  -1  -1  -1  -1  -1  -1  -1  2   -1  -1  -1  -1
5   -1  2   1   2   3   0   -1  2   -1  4   5   -1  -1  -1  4   -1  -1  -1  -1
1   -1  -1  2   -1  -1  -1  0   2   -1  2   -1  -1  -1  3   -1  5   -1  -1  -1
-1  -1  -1  -1  -1  -1  2   2   0   5   4   3   -1  -1  -1  -1  -1  1   -1  4
-1  -1  1   -1  -1  -1  -1  -1  5   0   -1  -1  3   -1  -1  -1  -1  -1  -1  -1
-1  -1  -1  3   -1  -1  4   2   4   -1  0   -1  4   2   -1  -1  -1  5   -1  2
-1  -1  -1  -1  2   -1  5   -1  3   -1  -1  0   -1  -1  -1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1  -1  -1  -1  3   4   -1  0   -1  1   1   1   3   4   -1
2   5   -1  -1  -1  -1  -1  -1  -1  -1  2   -1  -1  0   -1  -1  -1  5   -1  4
-1  -1  -1  -1  -1  -1  -1  3   -1  -1  -1  -1  1   -1  0   -1  -1  5   -1  5
-1  -1  5   -1  -1  2   4   -1  -1  -1  -1  -1  1   -1  -1  0   -1  -1  -1  -1
-1  2   -1  -1  1   -1  -1  5   -1  -1  -1  -1  1   -1  -1  -1  0   -1  -1  5
4   -1  -1  -1  1   -1  -1  -1  1   -1  5   -1  3   5   5   -1  -1  0   1   -1
5   2   -1  3   -1  -1  -1  -1  -1  -1  -1  -1  4   -1  -1  -1  -1  1   0   -1
4   2   -1  -1  5   -1  -1  -1  4   -1  2   -1  -1  4   5   -1  5   -1  -1  0

-----------------------------------
CS542 Link State Routing Simulator

(1) Create a Network Topology
(2) Build a Connection Table
(3) Shortest Path to Destination Router
(4) Modify a Topology
(5) Best Router for Broadcast
(6) Exit
```

```
Command: 2

Select a source router : 1
Router 1 Connection Table
Destination     Interface
========================
1               -
2               20
3               8
4               8
5               8
6               8
7               8
8               8
9               8
10              8
11              8
12              8
13              8
14              14
15              8
16              8
17              8
18              8
19              8
20              20

-----------------------------------
CS542 Link State Routing Simulator

(1) Create a Network Topology
(2) Build a Connection Table
(3) Shortest Path to Destination Router
(4) Modify a Topology
(5) Best Router for Broadcast
(6) Exit

Command: 3

Select a destination router : 3
Shortest Path from router 1 to router 3 is -
1 -> 8 -> 4 -> 7 -> 3
Cost = 6

-----------------------------------
CS542 Link State Routing Simulator

(1) Create a Network Topology
```

```
(2) Build a Connection Table
(3) Shortest Path to Destination Router
(4) Modify a Topology
(5) Best Router for Broadcast
(6) Exit

Command: 4

Select Router to be removed : 4
Router 1 Connection Table
Destination     Interface
=======================
1               -
2               20
3               8
4               -
5               8
6               8
7               8
8               8
9               8
10              8
11              8
12              8
13              8
14              14
15              8
16              8
17              8
18              8
19              8
20              20

Shortest Path from router 1 to router 3 is -
1 -> 8 -> 9 -> 7 -> 3
Cost = 7

-----------------------------------
CS542 Link State Routing Simulator

(1) Create a Network Topology
(2) Build a Connection Table
(3) Shortest Path to Destination Router
(4) Modify a Topology
(5) Best Router for Broadcast
(6) Exit

Command: 5
```

```
Broadcast Router is Router 5
Broadcast Cost (sum of path costs to all other routers) = 61
Router 5 Connection Table
Destination     Interface
========================
1               18
2               17
3               7
4               -
5               -
6               17
7               7
8               18
9               18
10              7
11              18
12              12
13              17
14              18
15              17
16              17
17              17
18              18
19              18
20              17
```

Results obtained for a network with 25 routers is as follows –

```
-----------------------------------
CS542 Link State Routing Simulator

(1) Create a Network Topology
(2) Build a Connection Table
(3) Shortest Path to Destination Router
(4) Modify a Topology
(5) Best Router for Broadcast
(6) Exit

Command: 1

Input original network topology matrix data file: matrix_25.txt
Review original topology matrix:
0   -1  -1  -1  4   -1  -1  -1  -1  5   5   5   -1  -1  -1  -1  -1  -1  -1  3   -1  -1  -1  -1  -1
-1  0   3   3   1   -1  -1  1   -1  5   -1  -1  -1  1   2   -1  -1  -1  -1  5   -1  -1  -1  -1  -1
-1  3   0   -1  -1  3   -1  -1  4   -1  -1  2   5   2   -1  -1  -1  -1  1   2   -1  -1  -1  -1  -1
-1  3   -1  0   -1  -1  -1  -1  5   -1  -1  -1  -1  -1  -1  5   -1  3   3   -1  -1  -1  -1  -1  -1
4   1   -1  -1  0   4   4   -1  -1  4   -1  1   -1  -1  -1  1   -1  -1  -1  -1  -1  2   -1  -1  -1
-1  -1  3   -1  4   0   -1  -1  4   -1  -1  -1  4   2   -1  -1  1   -1  -1  -1  -1  1   -1  -1  5
-1  -1  -1  -1  4   -1  0   2   5   3   -1  2   -1  -1  -1  -1  -1  -1  -1  -1  1   -1  -1  -1  -1
```

```
-1   1   -1   -1   -1   -1   2   0   3   4   -1   -1   5   -1   4   1   -1   -1   -1   -1   -1   1   -1   -1   4
-1   -1   4   5   -1   4   5   3   0   -1   -1   -1   1   -1   3   -1   -1   -1   -1   3   1   -1   -1   2   -1
5   5   -1   -1   4   -1   3   4   -1   0   -1   -1   4   -1   4   -1   -1   -1   4   -1   -1   -1   -1   -1   5
5   -1   -1   -1   -1   -1   -1   -1   -1   -1   0   -1   -1   -1   -1   -1   1   -1   1   -1   -1   -1   3   5   4
5   -1   2   -1   1   -1   2   -1   -1   -1   -1   0   1   -1   -1   -1   4   -1   -1   -1   -1   -1   2   -1   -1
-1   -1   5   -1   -1   4   -1   5   1   4   -1   1   0   -1   1   -1   -1   -1   -1   -1   -1   -1   3   -1   -1
-1   1   2   -1   -1   2   -1   -1   -1   -1   -1   -1   -1   0   -1   -1   -1   2   -1   -1   -1   -1   -1   1   2
-1   2   -1   -1   -1   -1   -1   4   3   4   -1   -1   1   -1   0   -1   3   -1   -1   4   -1   -1   -1   -1   3
-1   -1   -1   5   1   -1   -1   1   -1   -1   -1   -1   -1   -1   -1   -1   0   -1   -1   -1   2   -1   -1   -1   -1   -1
-1   -1   -1   -1   -1   1   -1   -1   -1   -1   1   4   -1   -1   3   -1   0   -1   -1   3   4   -1   -1   -1   -1
-1   -1   -1   3   -1   -1   -1   -1   -1   -1   -1   -1   -1   2   -1   -1   -1   0   -1   -1   -1   4   -1   -1   4
-1   -1   1   3   -1   -1   -1   -1   -1   4   1   -1   -1   -1   -1   -1   -1   -1   0   -1   -1   -1   -1   -1   -1
3   5   2   -1   -1   -1   -1   -1   3   -1   -1   -1   -1   -1   4   2   3   -1   -1   0   -1   -1   -1   -1   -1
-1   -1   -1   -1   -1   -1   1   -1   1   -1   -1   -1   -1   -1   -1   -1   4   -1   -1   -1   0   -1   3   3   4
-1   -1   -1   -1   2   1   -1   1   -1   -1   -1   -1   -1   -1   -1   -1   -1   4   -1   -1   -1   0   3   -1   -1
-1   -1   -1   -1   -1   -1   -1   -1   -1   -1   3   2   3   -1   -1   -1   -1   -1   -1   -1   3   3   0   -1   -1
-1   -1   -1   -1   -1   -1   -1   -1   2   -1   5   -1   -1   1   -1   -1   -1   -1   -1   -1   3   -1   -1   0   -1
-1   -1   -1   -1   -1   5   -1   4   -1   5   4   -1   -1   2   3   -1   -1   4   -1   -1   4   -1   -1   -1   0
```

```
-----------------------------------
CS542 Link State Routing Simulator

(1) Create a Network Topology
(2) Build a Connection Table
(3) Shortest Path to Destination Router
(4) Modify a Topology
(5) Best Router for Broadcast
(6) Exit

Command: 2

Select a source router : 12
Router 12 Connection Table
Destination     Interface
========================
1               5
2               5
3               3
4               5
5               5
6               5
7               7
8               5
9               13
10              7
11              3
12              -
13              13
14              5
```

```
15              13
16              5
17              17
18              5
19              3
20              5
21              13
22              5
23              23
24              5
25              5

-----------------------------------
CS542 Link State Routing Simulator

(1) Create a Network Topology
(2) Build a Connection Table
(3) Shortest Path to Destination Router
(4) Modify a Topology
(5) Best Router for Broadcast
(6) Exit

Command: 3

Select a destination router : 11
Shortest Path from router 12 to router 11 is -
12 -> 3 -> 19 -> 11
Cost = 4

-----------------------------------
CS542 Link State Routing Simulator

(1) Create a Network Topology
(2) Build a Connection Table
(3) Shortest Path to Destination Router
(4) Modify a Topology
(5) Best Router for Broadcast
(6) Exit

Command: 4

Select Router to be removed : 3
Router 12 Connection Table
Destination     Interface
========================
1               5
2               5
3               -
4               5
```

```
5               5
6               5
7               7
8               5
9               13
10              7
11              17
12              -
13              13
14              5
15              13
16              5
17              17
18              5
19              17
20              5
21              13
22              5
23              23
24              5
25              5

Shortest Path from router 12 to router 11 is -
12 -> 17 -> 11
Cost = 5

-----------------------------------
CS542 Link State Routing Simulator

(1) Create a Network Topology
(2) Build a Connection Table
(3) Shortest Path to Destination Router
(4) Modify a Topology
(5) Best Router for Broadcast
(6) Exit

Command: 5

Broadcast Router is Router 5
Broadcast Cost (sum of path costs to all other routers) = 71
Router 5 Connection Table
Destination     Interface
========================
1               1
2               2
3               -
4               2
5               -
6               22
```

7	12
8	16
9	12
10	10
11	22
12	12
13	12
14	2
15	12
16	16
17	22
18	2
19	22
20	16
21	12
22	22
23	12
24	2
25	2

Results obtained for a network with 30 routers is as follows –

```
-----------------------------------
CS542 Link State Routing Simulator

(1) Create a Network Topology
(2) Build a Connection Table
(3) Shortest Path to Destination Router
(4) Modify a Topology
(5) Best Router for Broadcast
(6) Exit

Command: 1

Input original network topology matrix data file: matrix_30.txt
Review original topology matrix:
0  1  -1  -1  -1  1  -1  -1  -1  2  -1  -1  -1  4  -1  -1  -1  -1  3  -1  3  5  1  2  -1  5  -1  -1  5  -1

1  0  1  4  5  -1  2  -1  5  -1  -1  2  -1  -1  -1  3  -1  -1  -1  -1  -1  -1  -1  5  3  -1  1  -1  -1  1

-1  1  0  -1  -1  1  -1  -1  -1  -1  3  -1  -1  -1  -1  -1  4  -1  -1  4  -1  1  4  2  -1  -1  -1  -1  1  -1

-1  4  -1  0  -1  -1  -1  5  -1  2  -1  -1  -1  -1  4  -1  -1  -1  -1  -1  5  5  2  -1  1  -1  -1  5  4  -1

-1  5  -1  -1  0  -1  -1  -1  -1  2  2  -1  -1  -1  2  3  -1  4  2  -1  -1  -1  -1  -1  -1  -1  2  4  -1  -1

1  -1  1  -1  -1  0  1  -1  -1  -1  3  5  5  -1  -1  5  -1  -1  1  -1  1  5  1  -1  5  -1  4  -1  -1  -1

-1  2  -1  -1  -1  1  0  -1  -1  -1  -1  -1  -1  -1  -1  -1  5  -1  -1  -1  -1  4  3  -1  -1  -1  -1  -1  -1  -1

-1  -1  -1  5  -1  -1  -1  0  -1  -1  5  5  3  -1  -1  -1  -1  -1  -1  -1  1  -1  -1  -1  2  -1  1  -1  4  -1

-1  5  -1  -1  -1  -1  -1  -1  0  1  5  -1  -1  -1  4  -1  -1  -1  -1  -1  -1  3  -1  -1  -1  -1  -1  5  -1  -1
```

```
2  -1  -1  2  2  -1  -1  -1  1  0  -1  -1  -1  2  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1  3  2  -1  -1  -1
-1  -1  3  -1  2  3  -1  5  5  -1  0  -1  -1  -1  -1  -1  -1  5  -1  -1  -1  -1  5  -1  5  -1  4  -1  -1  -1
-1  2  -1  -1  -1  5  -1  5  -1  -1  -1  0  -1  3  -1  -1  -1  -1  -1  4  -1  5  -1  -1  -1  2  -1  -1  3  4
-1  -1  -1  -1  -1  5  -1  3  -1  -1  -1  -1  0  1  1  -1  -1  -1  -1  -1  -1  5  4  -1  4  1  -1  -1  2  -1
4  -1  -1  -1  -1  -1  -1  -1  -1  2  -1  3  1  0  5  1  -1  -1  3  3  5  -1  -1  -1  5  -1  -1  -1  -1  -1
-1  -1  -1  4  2  -1  -1  -1  4  -1  -1  -1  1  5  0  -1  -1  -1  -1  4  -1  -1  -1  -1  -1  -1  5  -1  -1  -1
-1  3  -1  -1  3  5  -1  -1  -1  -1  -1  -1  -1  1  -1  0  -1  -1  4  5  -1  -1  4  -1  -1  5  -1  3  -1  -1
-1  -1  4  -1  -1  -1  5  -1  -1  -1  -1  -1  -1  -1  -1  -1  0  5  -1  5  -1  -1  -1  4  3  -1  2  -1  3  -1
-1  -1  -1  -1  4  -1  -1  -1  -1  -1  5  -1  -1  -1  -1  -1  5  0  -1  3  -1  -1  -1  -1  -1  1  1  -1  -1  4
3  -1  -1  -1  2  1  -1  -1  -1  -1  -1  -1  -1  3  -1  4  -1  -1  0  1  5  -1  -1  -1  -1  -1  -1  4  -1  3
-1  -1  4  -1  -1  -1  -1  -1  -1  -1  -1  4  -1  3  4  5  5  3  1  0  -1  -1  3  1  2  -1  -1  -1  -1  4
3  -1  -1  5  -1  1  -1  1  -1  -1  -1  -1  -1  5  -1  -1  -1  -1  5  -1  0  5  -1  2  1  -1  -1  -1  -1  1
5  -1  1  5  -1  5  4  -1  3  -1  -1  5  5  -1  -1  -1  -1  -1  -1  -1  5  0  -1  3  -1  -1  5  4  4  5
1  -1  4  2  -1  1  3  -1  -1  -1  5  -1  4  -1  -1  4  -1  -1  -1  3  -1  -1  0  -1  -1  1  2  -1  -1  -1
2  5  2  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1  4  -1  -1  1  2  3  -1  0  -1  -1  -1  -1  -1  4
-1  3  -1  1  -1  5  -1  2  -1  -1  5  -1  4  5  -1  -1  3  -1  -1  2  1  -1  -1  -1  0  -1  -1  -1  -1  -1
5  -1  -1  -1  -1  -1  -1  -1  -1  3  -1  2  1  -1  -1  5  -1  1  -1  -1  -1  -1  1  -1  -1  0  -1  -1  4  2
-1  1  -1  -1  2  4  -1  1  -1  2  4  -1  -1  -1  5  -1  2  1  -1  -1  -1  5  2  -1  -1  -1  0  -1  -1  -1
-1  -1  -1  5  4  -1  -1  -1  5  -1  -1  -1  -1  -1  -1  3  -1  -1  4  -1  -1  4  -1  -1  -1  -1  -1  0  -1  -1
5  -1  1  4  -1  -1  -1  4  -1  -1  -1  3  2  -1  -1  -1  3  -1  -1  -1  -1  4  -1  -1  -1  4  -1  -1  0  -1
-1  1  -1  -1  -1  -1  -1  -1  -1  -1  -1  4  -1  -1  -1  -1  -1  4  3  4  1  5  -1  4  -1  2  -1  -1  -1  0
```

```
-------------------------------------
CS542 Link State Routing Simulator

(1) Create a Network Topology
(2) Build a Connection Table
(3) Shortest Path to Destination Router
(4) Modify a Topology
(5) Best Router for Broadcast
(6) Exit

Command: 2

Select a source router : 23
Router 23 Connection Table
Destination     Interface
========================
```

```
1               1
2               1
3               6
4               4
5               27
6               6
7               6
8               27
9               1
10              1
11              6
12              26
13              26
14              26
15              26
16              26
17              27
18              26
19              6
20              6
21              6
22              6
23              -
24              1
25              6
26              26
27              27
28              6
29              6
30              6

-----------------------------------
CS542 Link State Routing Simulator

(1) Create a Network Topology
(2) Build a Connection Table
(3) Shortest Path to Destination Router
(4) Modify a Topology
(5) Best Router for Broadcast
(6) Exit

Command: 3

Select a destination router : 22
Shortest Path from router 23 to router 22 is -
23 -> 6 -> 3 -> 22
Cost = 3

-----------------------------------
```

```
CS542 Link State Routing Simulator

(1) Create a Network Topology
(2) Build a Connection Table
(3) Shortest Path to Destination Router
(4) Modify a Topology
(5) Best Router for Broadcast
(6) Exit

Command: 4

Select Router to be removed : 6
Router 23 Connection Table
Destination     Interface
========================
1               1
2               1
3               1
4               4
5               27
6               -
7               7
8               27
9               1
10              1
11              11
12              26
13              26
14              26
15              26
16              26
17              27
18              26
19              20
20              20
21              1
22              1
23              -
24              1
25              4
26              26
27              27
28              26
29              1
30              1

Shortest Path from router 23 to router 22 is -
23 -> 1 -> 2 -> 3 -> 22
Cost = 4
```

```
-----------------------------------
CS542 Link State Routing Simulator

(1) Create a Network Topology
(2) Build a Connection Table
(3) Shortest Path to Destination Router
(4) Modify a Topology
(5) Best Router for Broadcast
(6) Exit

Command: 5

Broadcast Router is Router 2
Broadcast Cost (sum of path costs to all other routers) = 80
Router 2 Connection Table
Destination     Interface
========================
1               1
2               -
3               3
4               30
5               27
6               -
7               7
8               27
9               27
10              27
11              3
12              12
13              1
14              16
15              1
16              16
17              27
18              27
19              30
20              3
21              30
22              3
23              1
24              3
25              30
26              1
27              27
28              16
29              3
30              30
```

Results obtained for a network with 35 routers is as follows –

```
-----------------------------------
CS542 Link State Routing Simulator

(1) Create a Network Topology
(2) Build a Connection Table
(3) Shortest Path to Destination Router
(4) Modify a Topology
(5) Best Router for Broadcast
(6) Exit

Command: 1

Input original network topology matrix data file: matrix_35.txt
Review original topology matrix:
0   -1   -1   2   -1   -1   -1   -1   -1   1   2   -1   5   -1   2   -1   -1   -1   -1   -1   -1   5   4   5   -1   -1   -1   -1   1   -1
3   -1   -1   -1   -1
-1   0   -1   -1   3   3   -1   -1   -1   -1   -1   -1   -1   -1   -1   -1   -1   -1   4   -1   -1   -1   -1   -1   -1   -1   -1   -1   -1   -1   -1
-1   4   3   4   -1
-1   -1   0   -1   4   -1   5   -1   -1   -1   -1   5   -1   -1   -1   -1   -1   -1   1   -1   -1   -1   -1   -1   5   -1   -1   -1   -1   -1
-1   -1   -1   -1   -1
2   -1   -1   0   1   -1   3   -1   3   -1   1   5   -1   -1   -1   1   -1   -1   -1   -1   -1   1   3   -1   -1   -1   -1   3   -1   -1
-1   -1   4   -1   -1
-1   3   4   1   0   -1   -1   -1   -1   -1   -1   -1   -1   3   4   3   -1   -1   -1   4   -1   -1   2   -1   4   -1   -1   -1   -1   -1
-1   -1   -1   -1   5
-1   3   -1   -1   -1   0   -1   -1   -1   2   -1   4   -1   -1   -1   -1   -1   -1   1   -1   -1   -1   5   -1   2   -1   1   -1   -1   -1
-1   -1   -1   -1   -1
-1   -1   5   3   -1   -1   0   -1   -1   -1   -1   -1   -1   -1   -1   -1   -1   -1   -1   -1   3   1   -1   2   -1   -1   -1   -1   2   5   -1
2   -1   -1   -1   -1
-1   -1   -1   -1   -1   -1   -1   0   -1   2   3   -1   -1   4   -1   -1   -1   5   3   -1   -1   3   -1   -1   -1   -1   2   -1   2   -1
3   -1   -1   -1   -1
-1   -1   -1   3   -1   -1   -1   -1   0   -1   2   -1   -1   -1   -1   -1   -1   -1   3   -1   1   3   4   -1   -1   -1   -1   5   4   -1
-1   -1   2   -1   -1
1   -1   -1   -1   -1   2   -1   2   -1   0   -1   -1   -1   2   -1   -1   5   -1   4   -1   4   3   -1   -1   -1   -1   -1   -1   -1   -1
3   -1   -1   -1   3
2   -1   -1   1   -1   -1   -1   3   2   -1   0   -1   -1   -1   -1   -1   -1   -1   -1   -1   2   -1   -1   -1   -1   -1   -1   -1   -1   3   4
2   3   2   1   -1
-1   -1   5   5   -1   4   -1   -1   -1   -1   -1   0   5   -1   -1   -1   5   -1   -1   -1   3   -1   -1   -1   1   -1   -1   2   5   -1
-1   -1   3   4   2
5   -1   -1   -1   -1   -1   -1   -1   -1   -1   -1   5   0   1   -1   -1   -1   -1   -1   -1   3   -1   -1   -1   -1   -1   -1   -1   -1   -1
-1   -1   -1   -1   4
-1   -1   -1   -1   3   -1   -1   4   -1   2   -1   -1   1   0   -1   4   2   -1   -1   -1   -1   -1   -1   -1   -1   -1   -1   -1   3   5
-1   -1   -1   -1   -1
2   -1   -1   -1   4   -1   -1   -1   -1   -1   -1   -1   -1   -1   0   -1   -1   -1   5   3   -1   -1   2   -1   -1   5   -1   3   -1   4
-1   -1   -1   -1   -1
-1   -1   -1   1   3   -1   -1   -1   -1   -1   -1   -1   -1   4   -1   0   -1   5   2   -1   -1   3   4   -1   -1   2   4   -1   -1   -1
1   -1   -1   4   -1
-1   -1   -1   -1   -1   -1   -1   -1   -1   5   -1   5   -1   2   -1   -1   0   -1   -1   3   -1   -1   2   -1   -1   -1   -1   -1   -1   -1
1   1   1   -1   2
-1   4   -1   -1   -1   -1   -1   5   -1   -1   -1   -1   -1   -1   -1   -1   5   -1   0   -1   4   1   1   -1   -1   1   -1   -1   1   -1   2
-1   -1   -1   -1   -1
-1   -1   1   -1   -1   1   -1   3   3   4   -1   -1   -1   -1   5   2   -1   -1   0   -1   4   3   3   -1   -1   -1   3   -1   -1   5
-1   -1   -1   -1   -1
-1   -1   -1   -1   4   -1   3   -1   -1   -1   2   -1   -1   -1   3   -1   3   4   -1   0   -1   4   -1   -1   3   -1   3   -1   -1   3
4   -1   3   1   -1
-1   -1   -1   -1   -1   -1   1   -1   1   4   -1   3   3   -1   -1   -1   -1   1   4   -1   0   3   -1   -1   -1   -1   4   2   -1   -1
3   -1   -1   -1   1
```

```
5   -1   -1   1   -1   -1   -1   3   3   3   -1   -1   -1   -1   -1   3   -1   1   3   4   3   0   -1   -1   -1   1   -1   -1   -1   4
3   -1   -1   -1   -1
4   -1   -1   3   2   5   2   -1   4   -1   -1   -1   -1   -1   2   4   2   -1   3   -1   -1   -1   0   -1   1   -1   4   2   -1   -1
5   -1   -1   1   -1
5   -1   -1   -1   -1   -1   -1   -1   -1   -1   -1   -1   -1   -1   -1   -1   -1   -1   -1   -1   -1   -1   -1   -1   0   5   -1   -1   -1   -1   -1
-1   -1   -1   -1   4
-1   -1   5   -1   4   2   -1   -1   -1   -1   -1   1   -1   -1   -1   -1   -1   1   -1   3   -1   -1   1   5   0   3   -1   -1   -1   -1
-1   4   3   -1   -1
-1   -1   -1   -1   -1   -1   -1   -1   -1   -1   -1   -1   -1   -1   -1   5   2   -1   -1   -1   -1   -1   1   -1   -1   3   0   3   -1   -1   1
-1   -1   5   -1   -1
-1   -1   -1   -1   -1   1   -1   2   -1   -1   -1   -1   -1   -1   -1   4   -1   -1   3   3   4   -1   4   -1   -1   3   0   -1   -1   1
5   -1   -1   -1   -1
-1   -1   -1   3   -1   -1   2   -1   5   -1   -1   2   -1   -1   3   -1   -1   1   -1   -1   2   -1   2   -1   -1   -1   -1   0   -1   3
1   -1   -1   -1   2
1   -1   -1   -1   -1   -1   5   2   4   -1   3   5   -1   3   -1   -1   -1   -1   -1   -1   -1   -1   -1   -1   -1   -1   -1   -1   0   1
2   -1   1   3   3
-1   -1   -1   -1   -1   -1   -1   -1   -1   -1   4   -1   -1   5   4   -1   -1   2   5   3   -1   4   -1   -1   -1   1   1   3   1   0
-1   2   -1   -1   -1
3   -1   -1   -1   -1   -1   2   3   -1   3   2   -1   -1   -1   -1   1   1   -1   -1   4   3   3   5   -1   -1   -1   5   1   2   -1
0   -1   -1   4   -1
-1   4   -1   -1   -1   -1   -1   -1   -1   -1   3   -1   -1   -1   -1   -1   1   -1   -1   -1   -1   -1   -1   -1   4   -1   -1   -1   -1   2
-1   0   1   -1   -1
-1   3   -1   4   -1   -1   -1   -1   2   -1   2   3   -1   -1   -1   -1   1   -1   -1   3   -1   -1   -1   -1   3   5   -1   -1   1   -1
-1   1   0   -1   -1
-1   4   -1   -1   -1   -1   -1   -1   -1   -1   1   4   -1   -1   -1   4   -1   -1   -1   1   -1   -1   1   -1   -1   -1   -1   -1   3   -1
4   -1   -1   0   -1
-1   -1   -1   -1   5   -1   -1   -1   -1   3   -1   2   4   -1   -1   -1   2   -1   -1   -1   1   -1   -1   4   -1   -1   -1   2   3   -1
-1   -1   -1   -1   0
```

```
-----------------------------------
CS542 Link State Routing Simulator

(1) Create a Network Topology
(2) Build a Connection Table
(3) Shortest Path to Destination Router
(4) Modify a Topology
(5) Best Router for Broadcast
(6) Exit

Command: 2

Select a source router : 11
Router 11 Connection Table
Destination     Interface
========================
1                 1
2                 33
3                 4
4                 4
5                 4
6                 4
7                 9
8                 8
9                 9
10                1
```

```
11              -
12              34
13              33
14              33
15              34
16              4
17              33
18              4
19              4
20              34
21              9
22              4
23              34
24              1
25              34
26              4
27              33
28              31
29              33
30              33
31              31
32              33
33              33
34              34
35              9

-----------------------------------
CS542 Link State Routing Simulator

(1) Create a Network Topology
(2) Build a Connection Table
(3) Shortest Path to Destination Router
(4) Modify a Topology
(5) Best Router for Broadcast
(6) Exit

Command: 3

Select a destination router : 12
Shortest Path from router 11 to router 12 is -
11 -> 34 -> 23 -> 25 -> 12
Cost = 4

-----------------------------------
CS542 Link State Routing Simulator

(1) Create a Network Topology
(2) Build a Connection Table
(3) Shortest Path to Destination Router
```

```
(4) Modify a Topology
(5) Best Router for Broadcast
(6) Exit

Command: 4

Select Router to be removed : 34
Router 11 Connection Table
Destination     Interface
========================
1               1
2               33
3               4
4               4
5               4
6               4
7               9
8               8
9               9
10              1
11              -
12              4
13              33
14              33
15              1
16              4
17              33
18              4
19              4
20              20
21              9
22              4
23              4
24              1
25              4
26              4
27              33
28              31
29              33
30              33
31              31
32              33
33              33
34              -
35              9

Shortest Path from router 11 to router 12 is -
11 -> 4 -> 22 -> 18 -> 25 -> 12
Cost = 5
```

```
-------------------------------------
CS542 Link State Routing Simulator

(1) Create a Network Topology
(2) Build a Connection Table
(3) Shortest Path to Destination Router
(4) Modify a Topology
(5) Best Router for Broadcast
(6) Exit

Command: 5

Broadcast Router is Router 18
Broadcast Cost (sum of path costs to all other routers) = 99
Router 18 Connection Table
Destination     Interface
=======================
1               30
2               2
3               25
4               22
5               22
6               25
7               21
8               22
9               21
10              22
11              22
12              25
13              21
14              21
15              25
16              28
17              28
18              -
19              25
20              25
21              21
22              22
23              25
24              21
25              25
26              22
27              30
28              28
29              30
30              30
31              28
32              28
```

```
33              30
34              -
35              21
```

Link State Routing Protocol-

Link state routing protocol is used in packet switched networks to route data packets from source to destination in the most efficient manner. This protocol involves generation of connection tables for each router within the network. The connection table for a router maps each address of the other routers to a link to which the given router should forward a packet to, so as to minimize the cost of the traversal path to the destination.

Usually for the construction of the connection tables, a path optimization algorithm like Dijkstra's algorithm is used. The process involves calculation of shortest paths between all pairs of routers based in the link costs between router within the network and subsequently mapping the first hop router in each of the shortest paths to the connection table of the source router in each of the paths.

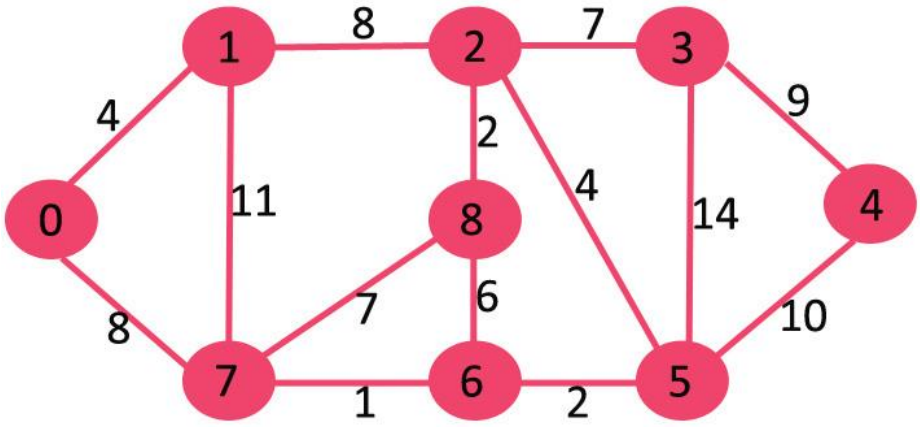
For example, if the shortest path from router 2 to router 6 in a network with 10 routers is 2-1-9-6, then in the connection table of 2, the 6th entry (corresponding to router 6 as the destination) would contain '1'.

That means, each router knows where to forward an incoming packet given the destination. The connection tables for all the routers in a network are updated following any change in the network topology or with the addition/removal of routers in the network.

Algorithm used to find shortest path-

In the given implementation, Dijkstra's algorithm was used to find the shortest path between source & destination routers in the network graph. It is illustrated below (illustration diagrams taken from geekforgeeks.org) –

Suppose we have the following network –

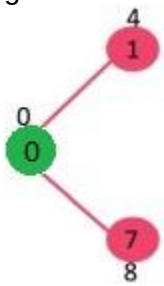


Let us assume that the source node is 0.

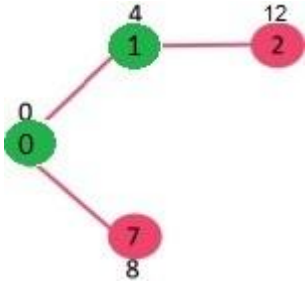
Our objective here, is to create a minimum spanning tree which spans across all the other nodes with the source node at the root.

We start off with an empty set $traversed$ which keeps a track of all the nodes included in the minimum spanning tree. Initially, we add the root node (in this case, node 0) to $traversed$ and initialize the path costs to all other nodes from the root node to *infinity*. The procedure cited below will run until the set $traversed$ includes all the nodes in the network.

1. The path costs to the neighboring nodes of the recently included node (0, for the first time) are updated. We get the following –

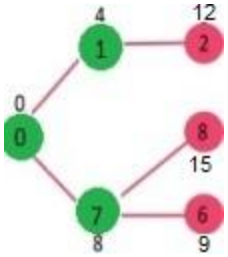


2. Next, the node in the graph having the minimum cost which has not already been included in $traversed$ is added to the set $traversed$ resulting in the following –

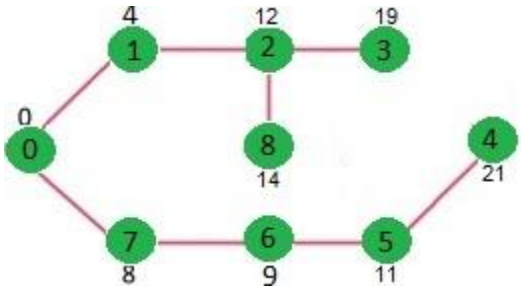


3. The path costs to the neighboring nodes of the recently added node (in this case, 4) are updated. Hence, the cost to node 2 becomes 12.

4. The same process is repeated again; that is, the node with the minimum path cost, not already included in $traversed$ is added to $traversed$ and the costs to the neighboring nodes of the newly added node are updated. The cost update function can simply be done by adding the path cost till the given node to the link costs between the given node and its adjacent nodes.



This process upon completion, results in a minimum spanning tree as illustrated below –



The shortest path from the source to any destination can be found out by traversing this minimum spanning tree from the root to the destination.

In this given implementation, a vector of lists is maintained as the graph is traversed. So, whenever a node is added to the $traversed$ set, the path traversed so far is also added as a list of nodes to the shortest path list corresponding to the source node (as a field in the Router object).

This process is repeated by varying the source node over every node in the graph. Hence, at the end, we obtain a list of shortest paths between any two nodes in the graph.

Compilation & running instructions-

The project was developed in C++ using Microsoft Visual Studio 2013. Therefore, Microsoft Visual C++ redistributable package is needed to run the executable on a windows machine.

The code can be compiled by running the following command in a developer command prompt after navigating to the directory containing all the source code files –

```
cl /c /Zi /nologo /W3 /WX- /O2 /Oi /Oy- /GL /D WIN32 /D NDEBUG /D _CONSOLE /D _LIB /D
_UNICODE /D UNICODE /Gm- /EHsc /MD /GS /Gy /Zc:wchar_t /Zc:forScope /Fo"Release\\"
/Fd"vc120.pdb" /Gd /TP /analyze- /errorReport:prompt main.cpp FileProcessor.cpp Network.cpp
Router.cpp
```

Here compiler used is cl.exe which is basically the Microsoft Visual C++ compiler.

The original Visual Studio solution with all the required build environments can be obtained here.

Following compilation, the program can be run simply by double clicking the resulting .exe file. Do keep in mind that the input files need to be in the same directory as the executable; otherwise, the whole path to the input file needs to be entered as a response to command 1 for loading the file.