

DATA STRUCTURES PROJECT

BY:

AMMAR SHERAZI (20I-0409)

MUHAMMAD DANIYAL (20i-0402)

AIZA (20i-0688)

DS-2001

Network Emulator:

The focus of this project is to emulate a network. There are 2 main classes designed that is:

1. Router
2. Graph

Graph:

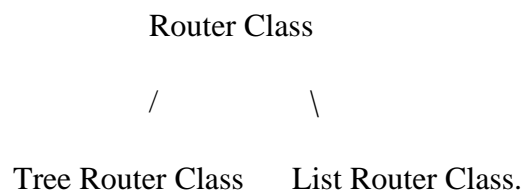
The graph class is design to link read and store from a csv file. It stores the adjacency matrix in a two-dimensional integer array. The array is then used by a global function (if called) to generate shortest paths from a source. The paths are stored and returned in a string list. This Graph class is crucial to generate first time data and a network.

```
class Graph {  
  
    int** adjMatrix;  
    int verticesCount;  
    int machinesCount;  
    int routersCount;  
  
public:  
    Graph() {  
        adjMatrix = nullptr;  
        verticesCount = 0;  
        routersCount = 0;  
        machinesCount = 0;  
    }  
};
```

Router:

This router class is divided into two other sub-classes, which are derived from the base router class itself.

A better representation would be:



Here the base router class is an abstract class. Which is then further divided into sub-classes that are used to make two different implantations of routers (Routing Tables of the router).

1. Using Linear List
2. Using Splay Tree

Simple structs that are used:

1. Message:

This struct stores all the messages fields, indicated by the requirements. The struct has every operator overloaded for templated uses.

2. RoutingTableField:

This struct stores the info for each row of a routing table. The struct has every operator overloaded for templated uses.

3. NamedQueue:

This struct is made to name every queue for the router.

Parser:

This is a simple parser used to detect commands and invalid commands and then execute respective driver functions according to the command entered.

Basic Data Structures (Templated):

1. Linked List
2. Queue
3. Max Heap (Incoming Queue)
4. Splay Tree

Command Line Interface & Query Format:

The program accepts following commands formats:

- send msg(to send a custom message)
- send msg filename(to send messages from file)
- print path M1 M2(to print path between two machines)
- print path M1* (to print path between machine to all others machines)

- print path* M1(to print path between from all others machines to a specific machine)
- change edge R1 M2 13 (to change an edge weight dynamically between two devices)
- print RT R1(to print a routing table)
- change RT R1 add M9 R4(to add a routing field to router, M9 = destination, R4 = next device)
- change RT R1 add filename(to add to routing tables `using` filename)change RT R1 remove M9 R4(to remove(`if` exists) a routing field from router, M9 = destination, R4 = next device)
- change RT R1 remove filename(to remove from routing tables `using` filename)
- cls(to clear the screen)
- exit(to exit program)