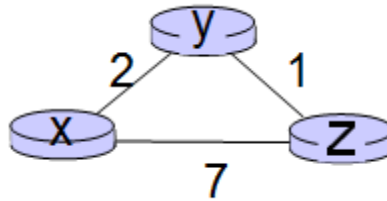


Lab 5 Distance Vector Routing Algorithm

Implement the Distance Vector routing algorithm using Java UDP socket API. You need to submit only ONE .java file to D2L.

THREE instances of your program will be run on the same machine to emulate three routers, e.g., X, Y, and Z. Each router/instance will read the network configuration from the *configuration.txt* file. In this file, the first line provides the port numbers that will be used in the three instances of your program. The other lines provide the weighted adjacency matrix of the network that can be visualized in the following figure.



The three routers/instances will exchange distance vector information between each other. After two rounds of message exchanging, the distance vector on each router will be stabilized.

For instance 1, your program should output something like this.

```
Enter the router's ID: X
Router X is running on port 6666
Distance vector on router X is:
<0, 2, 7>
Receives distance vector from router Y: <2, 0, 1>
Distance vector on router X is updated to:
<0, 2, 3>
Receives distance vector from router Z: <7, 1, 0>
Distance vector on router X is not updated
Receives distance vector from router Z: <3, 1, 0>
Distance vector on router X is not updated
```

For instance 2, your program should output something like this.

```
Enter the router's ID: Y
Router X is running on port 6667
Distance vector on router Y is:
<2, 0, 1>
Receives distance vector from router X: <0, 2, 7>
Distance vector on router Y is not updated
Receives distance vector from router Z: <7, 1, 0>
Distance vector on router Y is not updated
Receives distance vector from router X: <0, 2, 3>
Distance vector on router Y is not updated
Receives distance vector from router Z: <3, 1, 0>
Distance vector on router Y is not updated
```

For instance 3, your program should output something like this.

Enter the router's ID: Z

Router X is running on port 6668

Distance vector on router Z is:

<7, 1, 0>

Receives distance vector from router X: <0, 2, 7>

Distance vector on router Y is not updated

Receives distance vector from router Y: <2, 0, 1>

Distance vector on router Y is updated to:

<3, 1, 0>

Receives distance vector from router X: <0, 2, 3>

Distance vector on router Y is not updated