

CN ASSIGNMENT 4

Description:

This assignment is for implementing Distance-Vector-Routing protocol which follows a decentralized approach using multithreads. The file BT18CSE136_dvr.py contains the code. Outputs for two files (sample file mentioned in the instruction pdf sample.txt and a trial file trial.txt) are attached at the end of this report.

Working:

The main function will first read the input file, check the total number of routers and start a new thread for each router. Then it will wait for the thread to complete the execution. Each thread is considered as a router. It reads the input file and displays the initial status/routing table along with its neighbours name. Now the following actions will be done for the number of iterations given.

- Add the routing table in the shared queue for as many times as the number of its neighbour.
- Read the routing table of the neighbours in a loop. Stay there till the routing table of all neighbours is obtained.
- Using the Bellman Ford algorithm update the routing table.
- Wait for 0.5 seconds (for the ease of video recording)
- Display the updated table.

Testing:

The command line needs 2 arguments - python file and the input file. Failing to give either in the command line will raise an error.

```

C:\Users\Chotu Bacchu\Desktop>python BT18CSE136_dvr.py
Traceback (most recent call last):
  File "BT18CSE136_dvr.py", line 118, in <module>
    script, file = argv
ValueError: not enough values to unpack (expected 2, got 1)

C:\Users\Chotu Bacchu\Desktop>_

```

After writing the correct command on the terminal “python BT18CSE136_dvr.py filename.txt” the program will start running correctly. Here are the outputs of 2 routers (from 6).

```

C:\Users\Chotu Bacchu\Desktop>python BT18CSE136_dvr.py trial.txt
-----
Hi!! I am router E
meet my neighbours
  B D F
-----
Here is my initial status

E --> A = inf
E --> B = 3.0
E --> C = inf
E --> D = 3.0
E --> E = 0.0
E --> F = 6.0
-----

Hi!! I am router F
meet my neighbours
  A D E
-----
Here is my initial status

F --> A = 2.0
F --> B = inf
F --> C = inf
F --> D = 1.0
F --> E = 6.0
F --> F = 0.0
-----

```

After 1st iteration:

```

-----
Welcome again to router E
-----
This is the 1 iteration
My current status is

E --> A = 9.0***
E --> B = 3.0
E --> C = 5.0***
E --> D = 3.0
E --> E = 0.0
E --> F = 4.0***
-----

Welcome again to router F
-----
This is the 1 iteration
My current status is

F --> A = 2.0
F --> B = 4.0***
F --> C = 3.0***
F --> D = 1.0
F --> E = 4.0***
F --> F = 0.0
-----

```

After 2nd iteration :

```
-----
Welcome again to router F
-----
This is the 2 iteration
My current status is

F --> A = 2.0
F --> B = 4.0
F --> C = 3.0
F --> D = 1.0
F --> E = 4.0
F --> F = 0.0
-----

Welcome again to router E
-----
This is the 2 iteration
My current status is

E --> A = 6.0***
E --> B = 3.0
E --> C = 4.0***
E --> D = 3.0
E --> E = 0.0
E --> F = 4.0
-----
```

Since there was nothing to update in router F, no *** is present in the table. We can continue the process to as many iterations required.

Output:

For trial.txt file (from my side)

<https://drive.google.com/file/d/1eRSBqEkvy-sfcYxUuC84MuH57yEKghcr/view?usp=sharing>

For sample.txt file (in the instruction pdf)

<https://drive.google.com/file/d/1xwYsV-24-hX9JQrQrrKHj0xzSKA0FWpZ/view?usp=sharing>