BT19CSE028
Eshan Kumar Jain

CN Assignment - 4

Theory

A *distance-vector routing (DVR)* protocol requires that a router inform its neighbors of topology changes periodically. Historically known as the old ARPANET routing algorithm (or known as Bellman-Ford algorithm).

Bellman Ford Basics – Each router maintains a Distance Vector table containing the distance between itself and ALL possible destination nodes. Distances,based on a chosen metric, are computed using information from the neighbors' distance vectors.

Distance Vector Algorithm :–

1. A router transmits its distance vector to each of its neighbors in a routing packet.
2. Each router receives and saves the most recently received distance vector from each of its neighbors.
3. A router recalculates its distance vector when:
    - It receives a distance vector from a neighbor containing different information than before.
    - It discovers that a link to a neighbor has gone down.

The DV calculation is based on minimizing the cost to each destination

```
Dx(y) = Estimate of least cost from x to y

C(x,v) =  Node x knows cost to each neighbor v

Dx   =  [Dx(y): y ∈ N ] = Node x maintains distance vector

Node x also maintains its neighbors' distance vectors

– For each neighbor v, x maintains Dv = [Dv(y): y ∈ N ]
```

Multi-threading

Each router uses three separate threads for listening (for receiving distance-vector updates), sending (for sending distance-vector updates) and Bellman-Ford calculations (when link cost changes).

Working

Upon initialization, each router creates a distance-vector update packet and sends this packet to all direct neighbors. Upon receiving this distance-vector update packet, each neighboring router will incorporate the provided information into its routing table. Each router periodically broadcasts the distance-vector update packet to its neighbors.

On receiving distance-vector update packets from all other routers, a router builds up a reachability matrix. Given a view of the neighboring routers and their reachability, a router runs the Bellman-Ford algorithm to compute least-cost paths to all other routers within the network.

If the cost of a link changes, the connected routers recalculate the cost of reaching other routers and also provide an update to their neighbors, who will then notify their neighbors and so on until the network converges.

Instructions for Running the code

Code Files and Output Files are included in the Zip Folder sent along with Images of the Routing.

For First input file
python BT19CSE028_dvr.py input1.txt > routing_code1.txt

For Second input file
python BT19CSE028_dvr.py input2.txt > routing_code2.txt

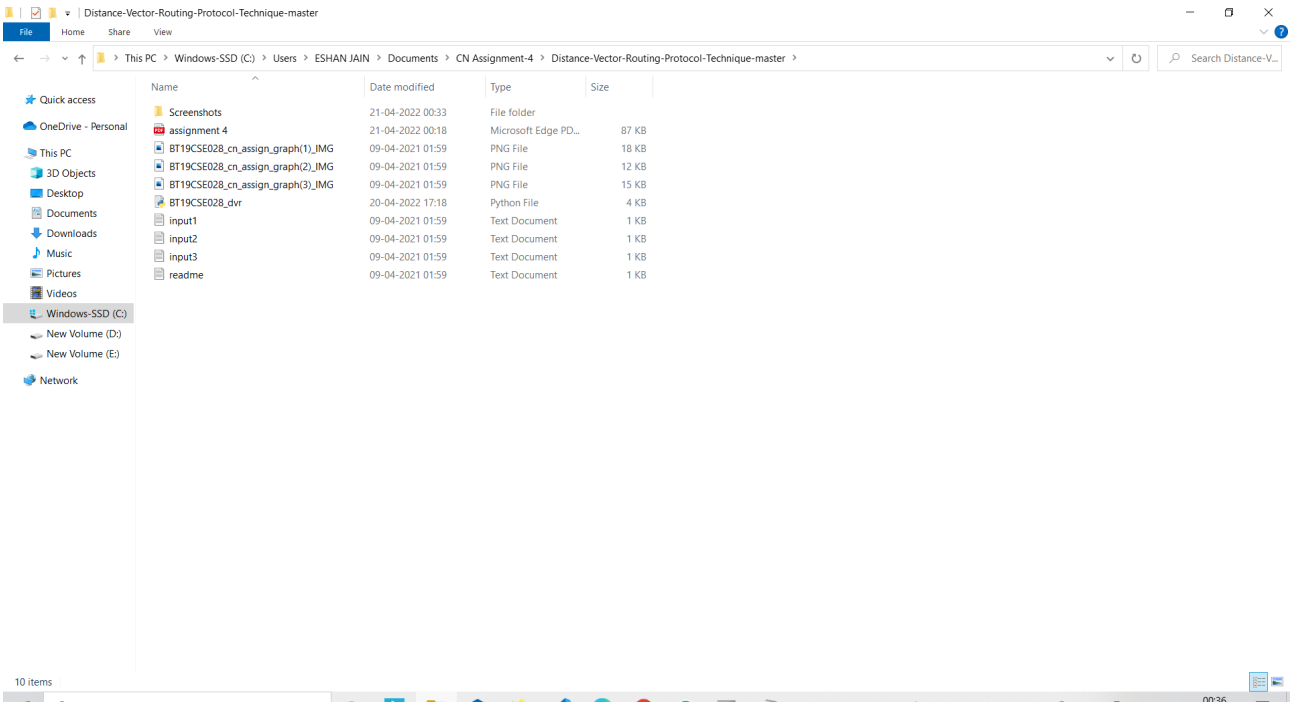
For Third input file
python BT19CSE028_dvr.py input3.txt > routing_code3.txt

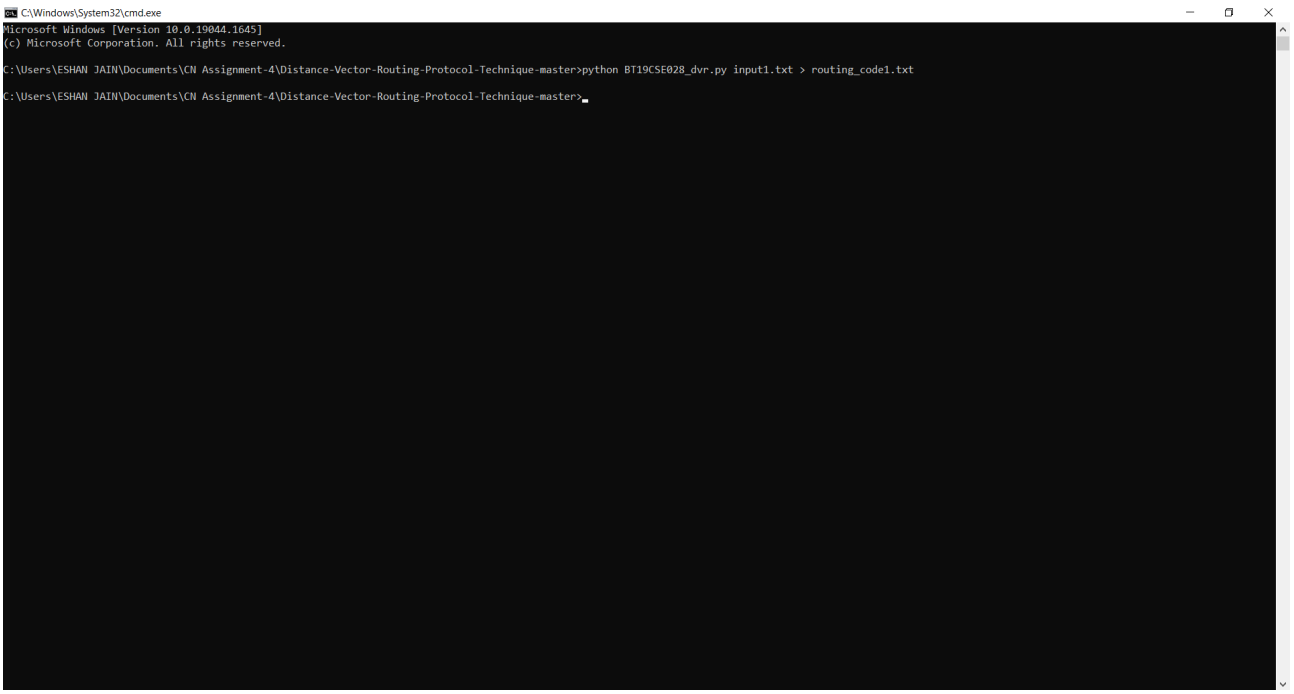
Testing of file 1 :-

Input1.txt

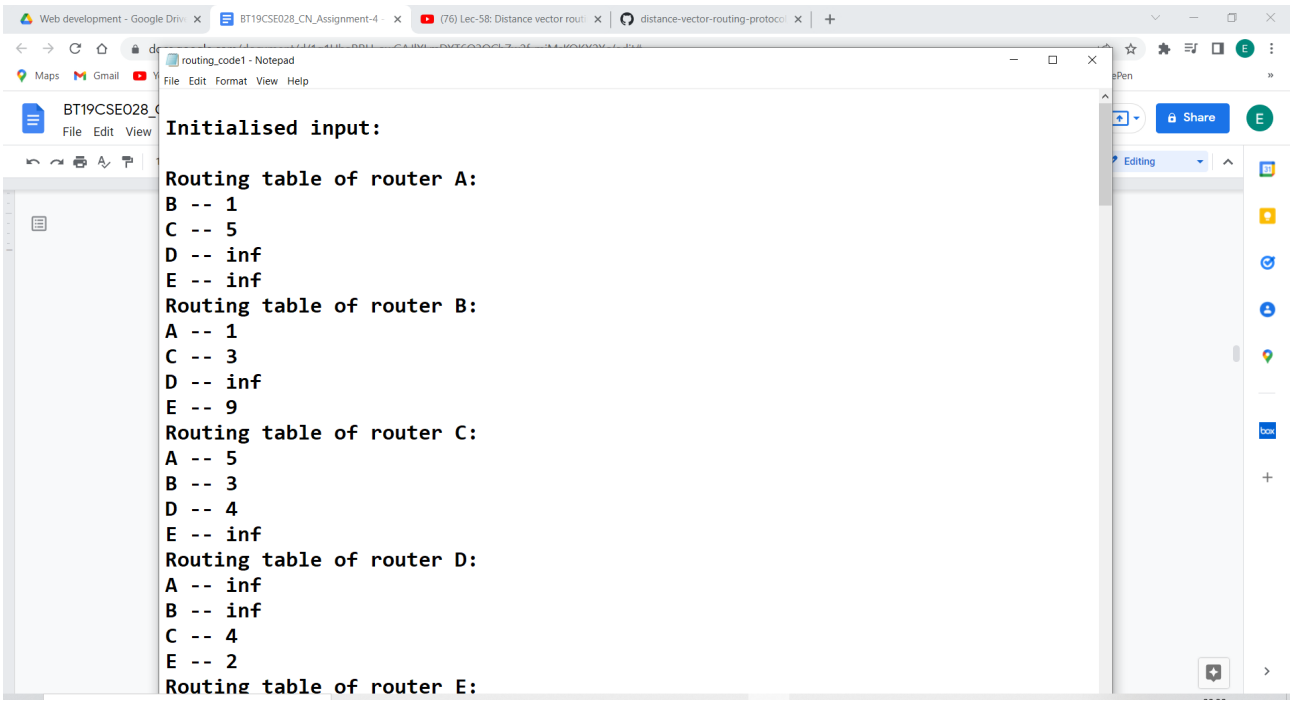
5
A B C D E
A B 1
A C 5
B C 3
C D 4
B E 9
D E 2
EOF

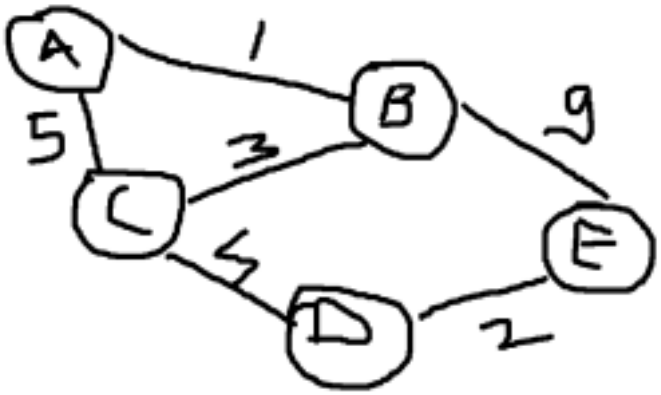
Before Running the Command…….





After Running the Command……..





Initialized input:

Routing table of router A:
B -- 1
C -- 5
D -- inf
E -- inf
Routing table of router B:
A -- 1
C -- 3
D -- inf
E -- 9

```
Routing table of router C:
A -- 5
B -- 3
D -- 4
E -- inf
Routing table of router D:
A -- inf
B -- inf
C -- 4
E -- 2
Routing table of router E:
A -- inf
B -- 9
C -- inf
D -- 2
================================================
==  Iteration 1
================================================

Routing table of router A with next hop:
        B -- 1 -- B
 * C -- 4 -- B
 * D -- 9 -- C
 * E -- 10 -- B
Routing table of router B with next hop:
        A -- 1 -- A
        C -- 3 -- C
 * D -- 7 -- C
        E -- 9 -- E
Routing table of router C with next hop:
 * A -- 4 -- B
        B -- 3 -- B
        D -- 4 -- D
 * E -- 6 -- D
Routing table of router D with next hop:
 * A -- 9 -- C
 * B -- 7 -- C
        C -- 4 -- C
        E -- 2 -- E
Routing table of router E with next hop:
 * A -- 10 -- B
        B -- 9 -- B
 * C -- 6 -- D
        D -- 2 -- D


================================================
==  Iteration 2
================================================
```

Routing table of router A with next hop:
        B -- 1 -- B
        C -- 4 -- B
 * D -- 8 -- B
        E -- 10 -- B
Routing table of router B with next hop:
        A -- 1 -- A
        C -- 3 -- C
        D -- 7 -- C
        E -- 9 -- E
Routing table of router C with next hop:
        A -- 4 -- B
        B -- 3 -- B
        D -- 4 -- D
        E -- 6 -- D
Routing table of router D with next hop:
 * A -- 8 -- C
        B -- 7 -- C
        C -- 4 -- C
        E -- 2 -- E
Routing table of router E with next hop:
        A -- 10 -- B
        B -- 9 -- B
        C -- 6 -- D
        D -- 2 -- D

==========
== Iteration 3
==========

Routing table of router A with next hop:
        B -- 1 -- B
        C -- 4 -- B
        D -- 8 -- B
        E -- 10 -- B
Routing table of router B with next hop:
        A -- 1 -- A
        C -- 3 -- C
        D -- 7 -- C
        E -- 9 -- E
Routing table of router C with next hop:
        A -- 4 -- B
        B -- 3 -- B
        D -- 4 -- D
        E -- 6 -- D
Routing table of router D with next hop:
        A -- 8 -- C
        B -- 7 -- C

C -- 4 -- C
E -- 2 -- E

Routing table of router E with next hop:
A -- 10 -- B
B -- 9 -- B
C -- 6 -- D
D -- 2 -- D

==========
== Iteration 4
==========

Routing table of router A with next hop:
B -- 1 -- B
C -- 4 -- B
D -- 8 -- B
E -- 10 -- B

Routing table of router B with next hop:
A -- 1 -- A
C -- 3 -- C
D -- 7 -- C
E -- 9 -- E

Routing table of router C with next hop:
A -- 4 -- B
B -- 3 -- B
D -- 4 -- D
E -- 6 -- D

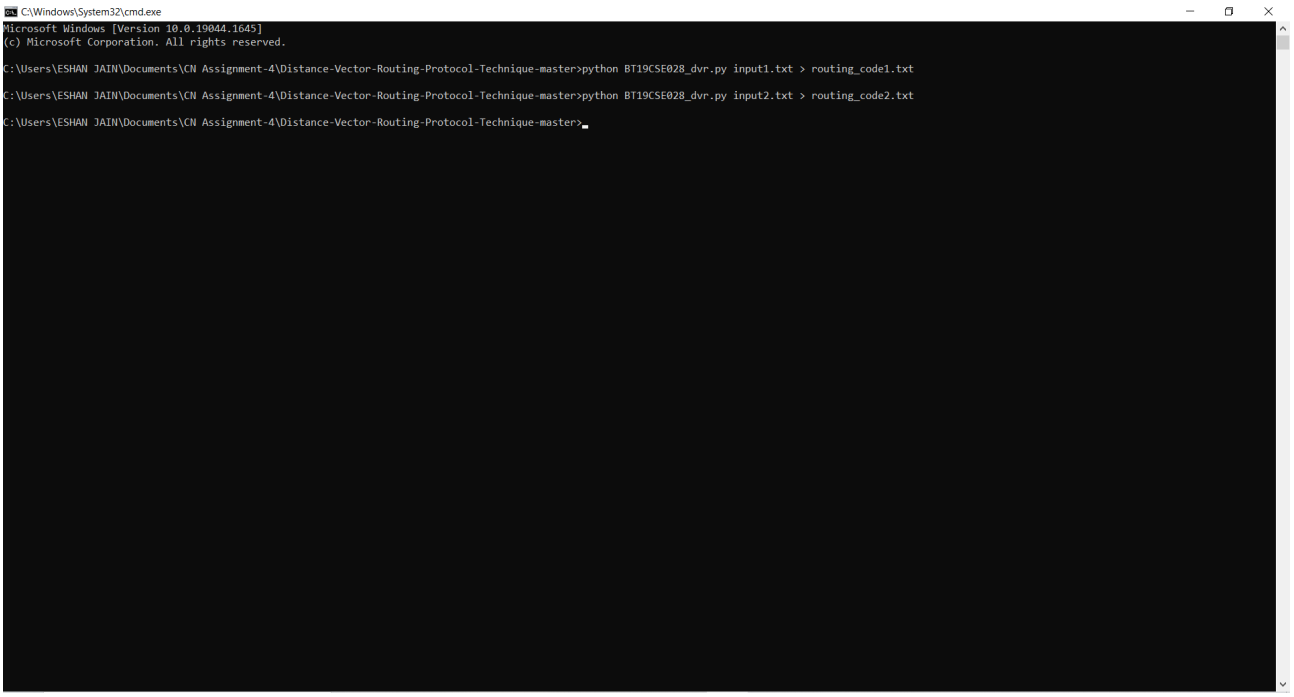
Routing table of router D with next hop:
A -- 8 -- C
B -- 7 -- C
C -- 4 -- C
E -- 2 -- E

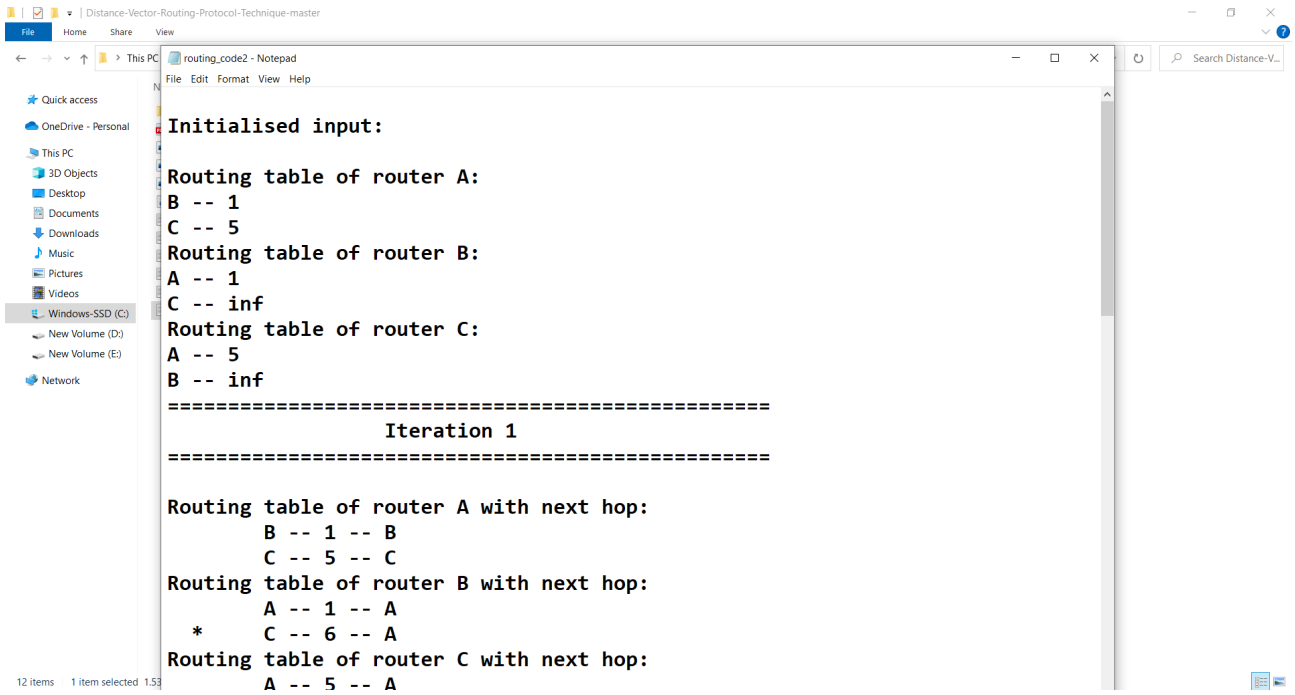
Routing table of router E with next hop:
A -- 10 -- B
B -- 9 -- B
C -- 6 -- D
D -- 2 – D

Testing of file 2 :-

Input2.txt

3
A B C
A B 1
A C 5
EOF





Initialized input:

Routing table of router A:
B -- 1
C -- 5
Routing table of router B:
A -- 1
C -- inf
Routing table of router C:
A -- 5
B -- inf
==
Iteration 1
==

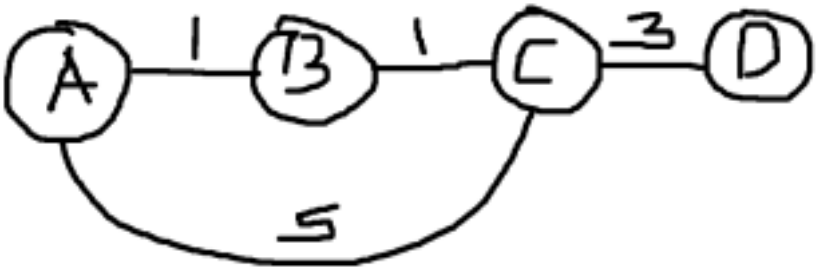
Routing table of router A with next hop:
B -- 1 -- B
C -- 5 -- C
Routing table of router B with next hop:
A -- 1 -- A
* C -- 6 -- A
Routing table of router C with next hop:
A -- 5 -- A
* B -- 6 -- A

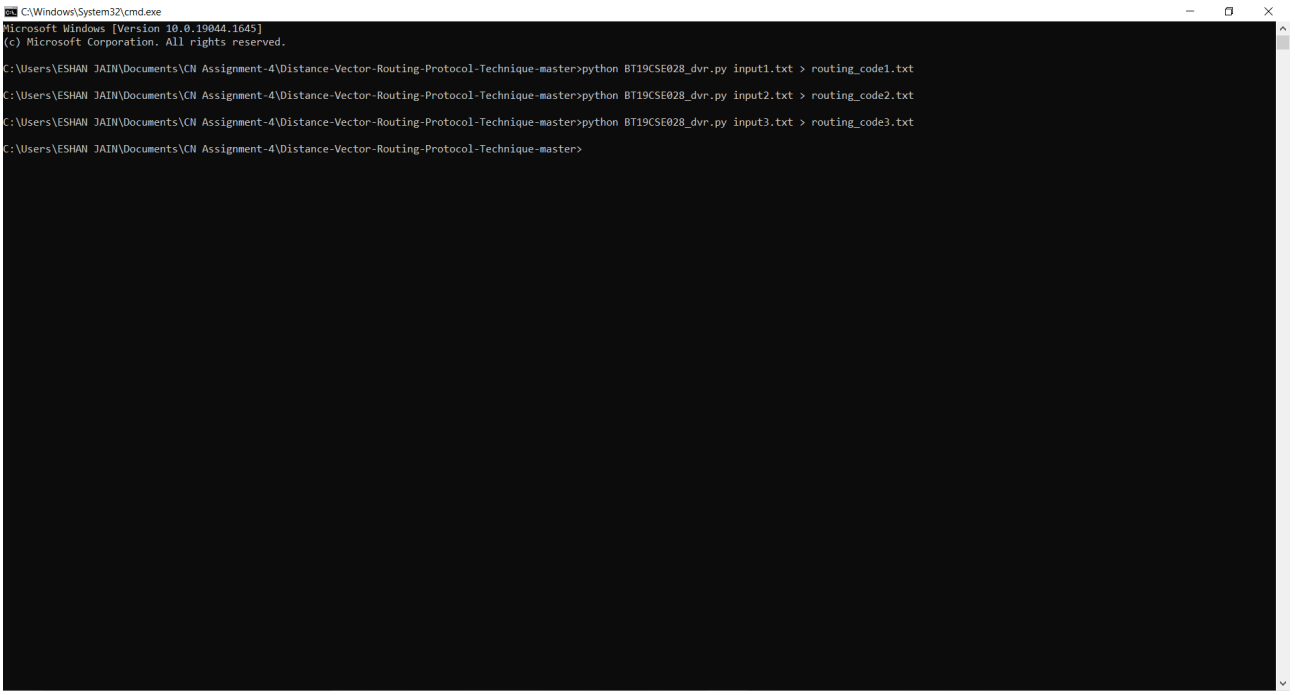
```
================================================
 Iteration 2
================================================

Routing table of router A with next hop:
        B -- 1 -- B
        C -- 5 -- C
Routing table of router B with next hop:
        A -- 1 -- A
        C -- 6 -- A
Routing table of router C with next hop:
        A -- 5 -- A
        B -- 6 -- A


================================================
==  Iteration 3
================================================

Routing table of router A with next hop:
        B -- 1 -- B
        C -- 5 -- C
Routing table of router B with next hop:
        A -- 1 -- A
        C -- 6 -- A
Routing table of router C with next hop:
        A -- 5 -- A
        B -- 6 -- A


================================================
==  Iteration 4
================================================

Routing table of router A with next hop:
        B -- 1 -- B
        C -- 5 -- C
Routing table of router B with next hop:
        A -- 1 -- A
        C -- 6 -- A
Routing table of router C with next hop:
        A -- 5 -- A
        B -- 6 -- A
```

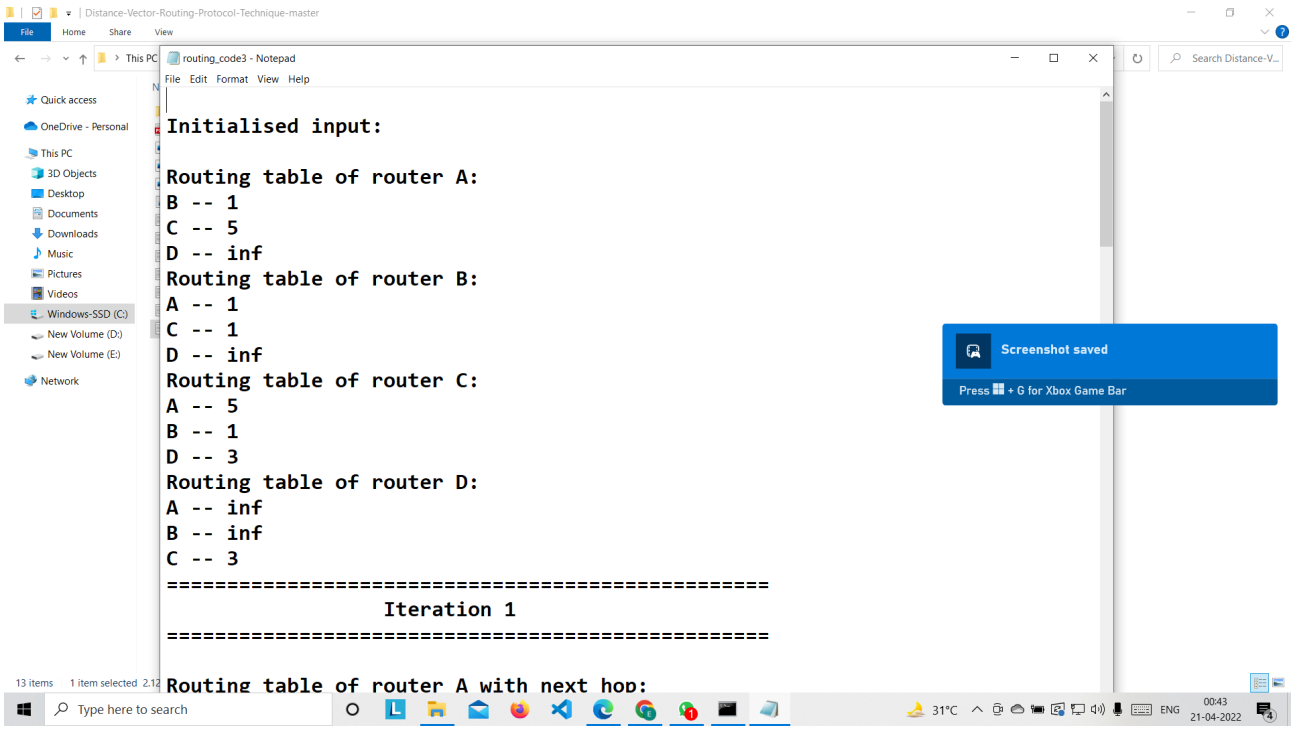
Testing of file 3 :-

Input3.txt

4
A B C D
A B 1
A C 5
B C 1
C D 3
EOF







Initialized input:

Routing table of router A:
B -- 1
C -- 5
D -- inf
Routing table of router B:
A -- 1
C -- 1
D -- inf
Routing table of router C:
A -- 5
B -- 1
D -- 3
Routing table of router D:
A -- inf
B -- inf
C -- 3
==========
== Iteration 1
==========

Routing table of router A with next hop:
B -- 1 -- B
* C -- 2 -- B

```
 * D -- 8 -- C
Routing table of router B with next hop:
        A -- 1 -- A
        C -- 1 -- C
 * D -- 4 -- C
Routing table of router C with next hop:
 * A -- 2 -- B
        B -- 1 -- B
        D -- 3 -- D
Routing table of router D with next hop:
 * A -- 8 -- C
 * B -- 4 -- C
        C -- 3 -- C


================================================
==  Iteration 2
================================================

Routing table of router A with next hop:
        B -- 1 -- B
        C -- 2 -- B
 * D -- 5 -- B
Routing table of router B with next hop:
        A -- 1 -- A
        C -- 1 -- C
        D -- 4 -- C
Routing table of router C with next hop:
        A -- 2 -- B
        B -- 1 -- B
        D -- 3 -- D
Routing table of router D with next hop:
 * A -- 5 -- C
        B -- 4 -- C
        C -- 3 -- C


================================================
==  Iteration 3
================================================

Routing table of router A with next hop:
        B -- 1 -- B
        C -- 2 -- B
        D -- 5 -- B
Routing table of router B with next hop:
```

A -- 1 -- A
C -- 1 -- C
D -- 4 -- C
Routing table of router C with next hop:
A -- 2 -- B
B -- 1 -- B
D -- 3 -- D
Routing table of router D with next hop:
A -- 5 -- C
B -- 4 -- C
C -- 3 -- C

==========
== Iteration 4
==========

Routing table of router A with next hop:
B -- 1 -- B
C -- 2 -- B
D -- 5 -- B
Routing table of router B with next hop:
A -- 1 -- A
C -- 1 -- C
D -- 4 -- C
Routing table of router C with next hop:
A -- 2 -- B
B -- 1 -- B
D -- 3 -- D
Routing table of router D with next hop:
A -- 5 -- C
B -- 4 -- C
C -- 3 -- C