COMPUTER NETWORKS (CSL 317)

Assignment 3 Report

NAME: DWEEJA REDDY DEVI

ENROLLMENT NUMBER: BT20CSE032

Problem Statement:

Implementation of Distance Vector Routing Protocol

Since it follows a decentralised approach, you will be implementing this algorithm via multithreading in Python. Each thread will represent a router instance. The threads (routers) can communicate through a shared queue.

Introduction and working of algorithm:

- ➢ Distance vector routing algorithm, also known as the Bellman-Ford algorithm, is a routing protocol used in computer networks to determine the best path for data to travel from one network node to another.

- ➢ In this algorithm, each node maintains a table that contains the distance to all known destinations and the next hop for each destination. The distance is typically measured in the number of hops, which is the number of routers that need to be crossed to reach the destination.

- ➢ The algorithm works by exchanging routing information between neighbouring nodes in the network. Each node sends its routing table to its neighbours, who then update their own routing tables. The distance to a destination is calculated by adding the cost of the link to the neighbour and the distance to the destination as advertised by the neighbour.

- ➢ This process continues until all nodes have converged on the best path to each destination. The algorithm is iterative and may take some time to converge, especially in large networks with many nodes.

- ➢ The major advantage of the distance vector routing algorithm is its simplicity and low overhead. However, it has some limitations, such as the inability to handle changes in the network topology quickly and efficiently, which can result in suboptimal routing paths.

Instructions to run the code:

Step 1 : Open the BT20CSE032_DVR.py file in any code editor (like vscode) and open the terminal.

Step 2 : Make sure there is an **inp.txt** file present in the same folder as the python file

Step 3 : Command to execute the file :

```
python  BT20CSE032_DVR.py inp.txt > out.txt
```

OR

```
python3 BT20CSE032_DVR.py inp.txt > out.txt
```

(Based on the python version that runs on your computer)

Step 4 : This will create a new file **out.txt** and the output(routing tables in every iteration) is saved in that file.

Code flow:

➢ The code takes an input file as an argument and reads the following from the input file:
  - Number of nodes in the network
  - Names of the nodes in the network
  - Edges and their weights in the network

➢ Each router is represented as an object of the `router` class, which has the following attributes:
  - `name`: name of the router

   - `id`: id of the router (computed from the name)
   - `fwd`: a dictionary containing the shortest distance to each node in the network from this router
   - `next_hop`: a list containing the next hop router for each node in the network
   - `neighbours`: a list of the neighbours of the router
   - `updated`: a list of nodes whose routing table was updated in the previous iteration

➢ The `Bellman_Ford` function updates the routing table of a single router using the Bellman-Ford algorithm. It takes two arguments: `routers` (a list of `router` objects) and `i` (the index of the router to be updated). The function updates the routing table of the router and appends the nodes whose routing table was updated to the `updated` list.

➢ The `Propagate` function is called by each thread and propagates the routing table of a router to its neighbours. It takes two arguments: `routers` (a list of `router` objects) and `i` (the index of the router whose routing table is being propagated). The function puts the routing table of the router in the queue of each neighbour, and waits until all neighbours have received the routing table. It then calls the `Bellman_Ford` function to update the routing table of the router, clears the queue, and sleeps for 2 seconds.

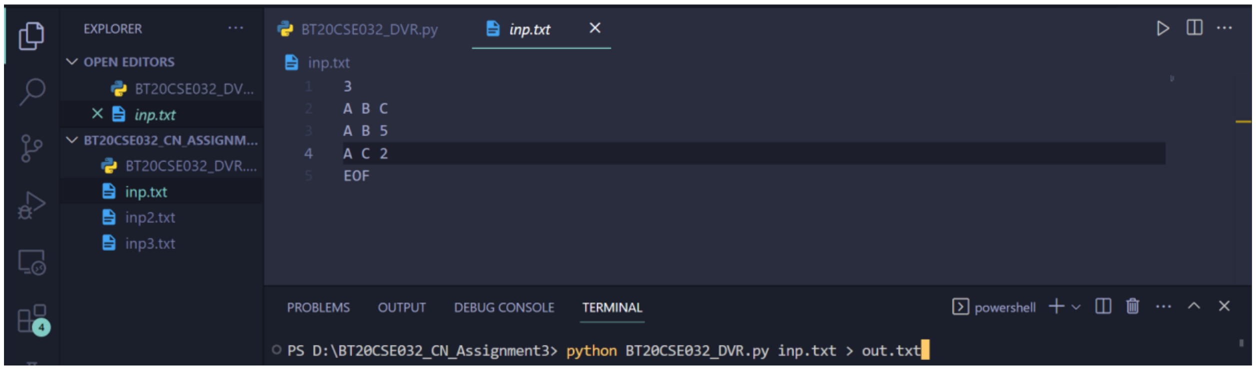
➢ The main function reads the input file and initialises the routers. It then creates a list of queues (`queueList`) with length equal to the number of nodes in the network. It creates four threads for each router and calls the `Propagate` function with the `routers` list and the index of the router as arguments. The main function then waits for all threads to complete, and prints the routing tables of each router at the end of each iteration.

➢ The code is designed to simulate the behaviour of routers in a network and update their routing tables. It is a distributed algorithm where each router computes its own routing table based on information received from its neighbours. The algorithm runs for a fixed number of iterations (4 in this case) to converge to a stable state where the routing tables do not change further.

Running the Test Cases:

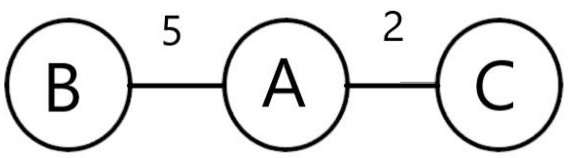
Test case 1:

Input:

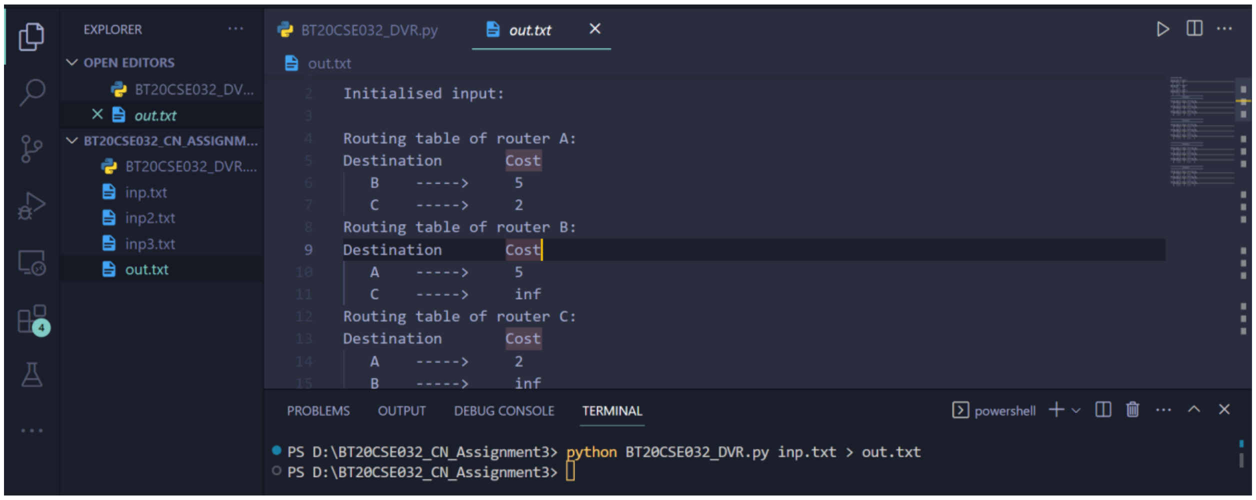


```
3
A B C
A B 5
A C 2
EOF
```

```
PS D:\BT20CSE032_CN_Assignment3> python BT20CSE032_DVR.py inp.txt > out.txt
```

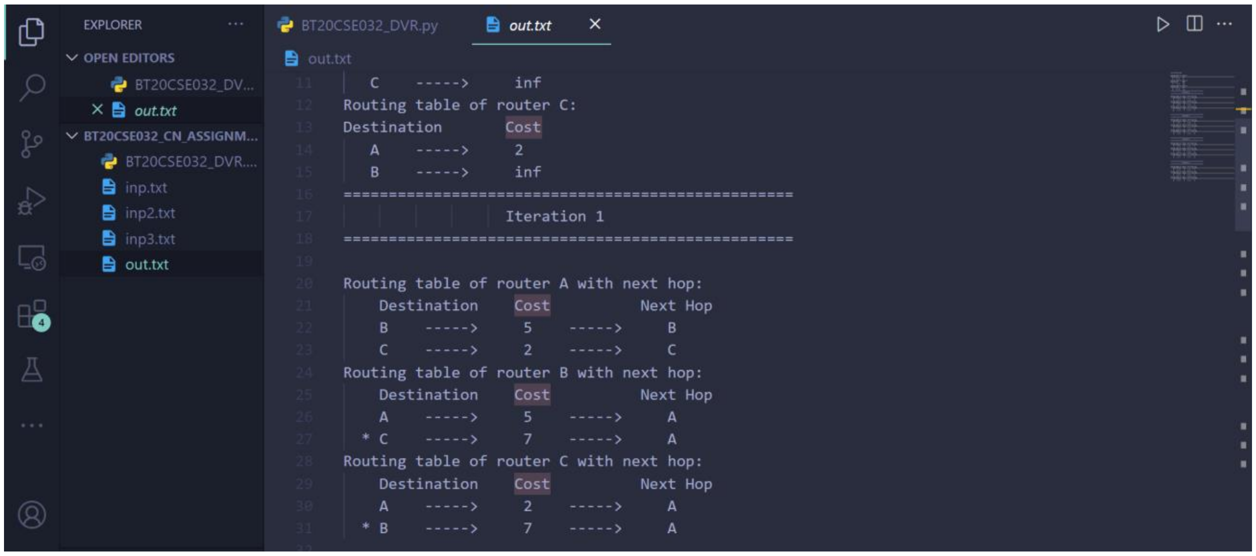


Output:

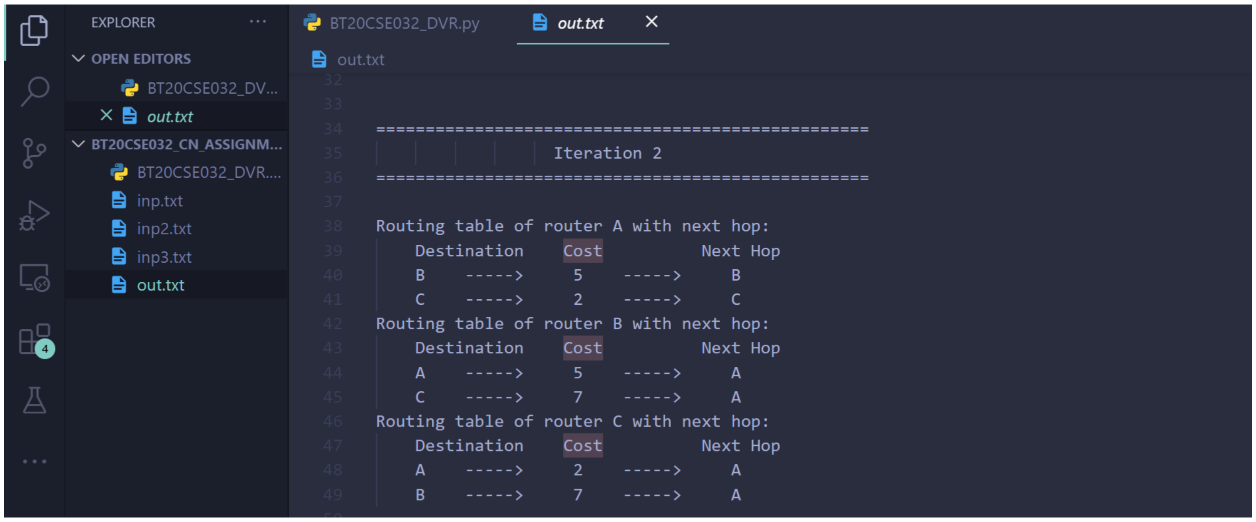


```
Initialised input:

Routing table of router A:
Destination        Cost
    B    ----->     5
    C    ----->     2
Routing table of router B:
Destination        Cost
    A    ----->     5
    C    ----->     inf
Routing table of router C:
Destination        Cost
    A    ----->     2
    B    ----->     inf
```

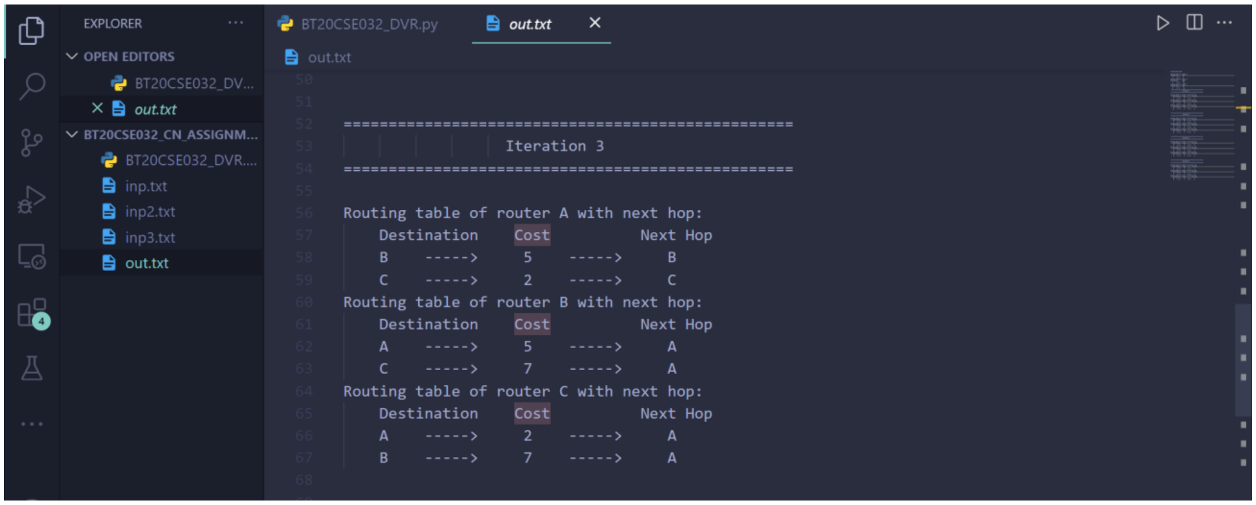
```
PS D:\BT20CSE032_CN_Assignment3> python BT20CSE032_DVR.py inp.txt > out.txt
PS D:\BT20CSE032_CN_Assignment3>
```



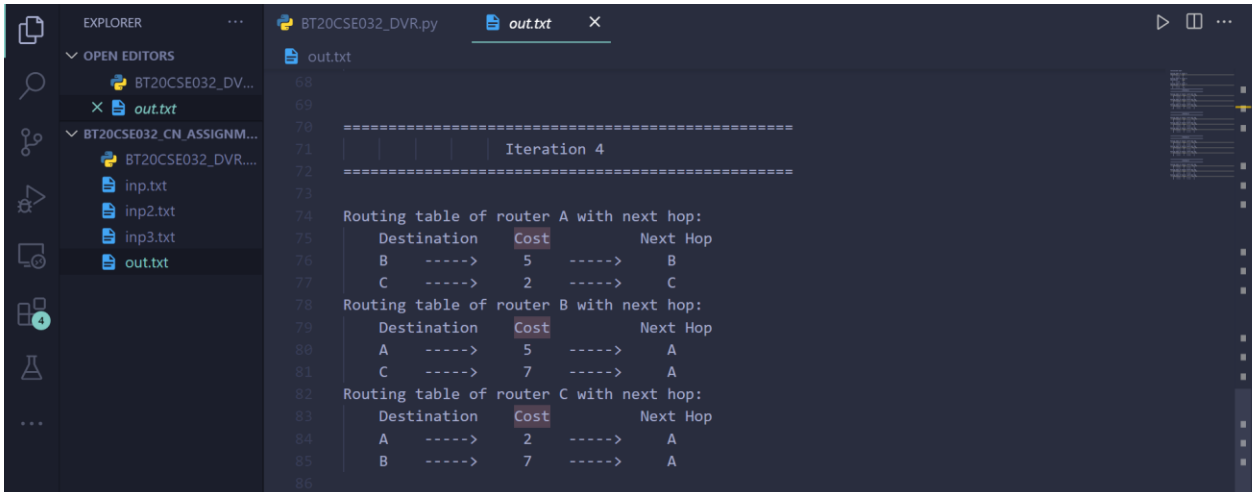
```
    C    ----->     inf
Routing table of router C:
Destination        Cost
    A    ----->     2
    B    ----->     inf
==================================================
                  Iteration 1
==================================================

Routing table of router A with next hop:
    Destination    Cost          Next Hop
     B    ----->     5    ----->     B
     C    ----->     2    ----->     C
Routing table of router B with next hop:
    Destination    Cost          Next Hop
     A    ----->     5    ----->     A
   * C    ----->     7    ----->     A
Routing table of router C with next hop:
    Destination    Cost          Next Hop
     A    ----->     2    ----->     A
   * B    ----->     7    ----->     A
```



```


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

Routing table of router A with next hop:
    Destination    Cost          Next Hop
     B    ----->     5    ----->     B
     C    ----->     2    ----->     C
Routing table of router B with next hop:
    Destination    Cost          Next Hop
     A    ----->     5    ----->     A
     C    ----->     7    ----->     A
Routing table of router C with next hop:
    Destination    Cost          Next Hop
     A    ----->     2    ----->     A
     B    ----->     7    ----->     A
```



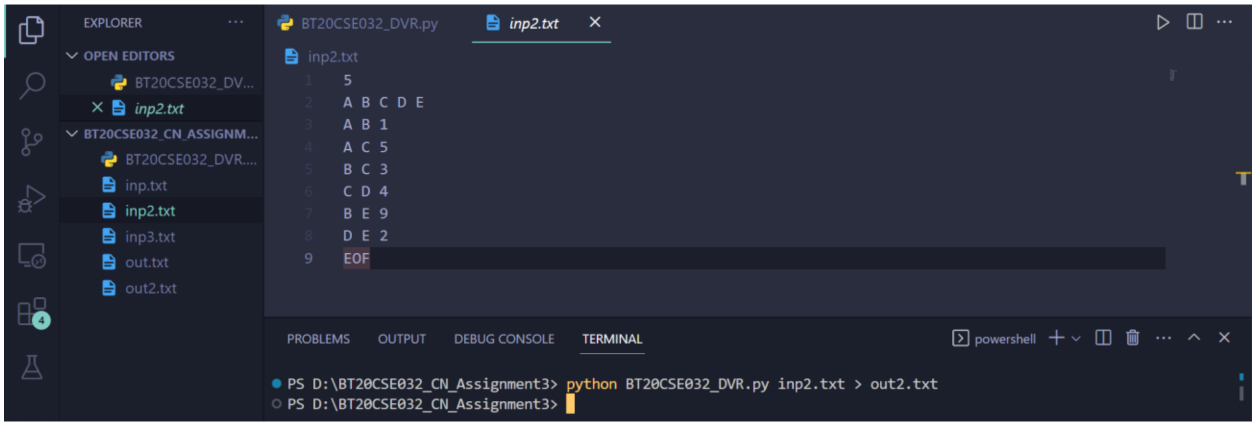
```
================================================
|    |    |    |    |    Iteration 3
================================================

Routing table of router A with next hop:
    Destination     Cost           Next Hop
    B     ----->     5     ----->     B
    C     ----->     2     ----->     C
Routing table of router B with next hop:
    Destination     Cost           Next Hop
    A     ----->     5     ----->     A
    C     ----->     7     ----->     A
Routing table of router C with next hop:
    Destination     Cost           Next Hop
    A     ----->     2     ----->     A
    B     ----->     7     ----->     A
```



```
================================================
|    |    |    |    |    Iteration 4
================================================

Routing table of router A with next hop:
    Destination     Cost           Next Hop
    B     ----->     5     ----->     B
    C     ----->     2     ----->     C
Routing table of router B with next hop:
    Destination     Cost           Next Hop
    A     ----->     5     ----->     A
    C     ----->     7     ----->     A
Routing table of router C with next hop:
    Destination     Cost           Next Hop
    A     ----->     2     ----->     A
    B     ----->     7     ----->     A
```

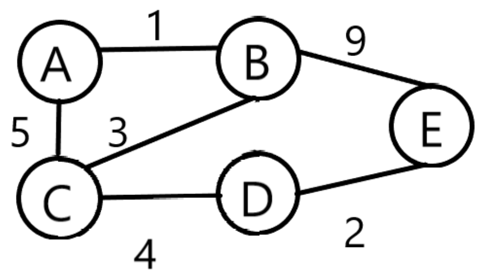
Test Case 2:

Input:

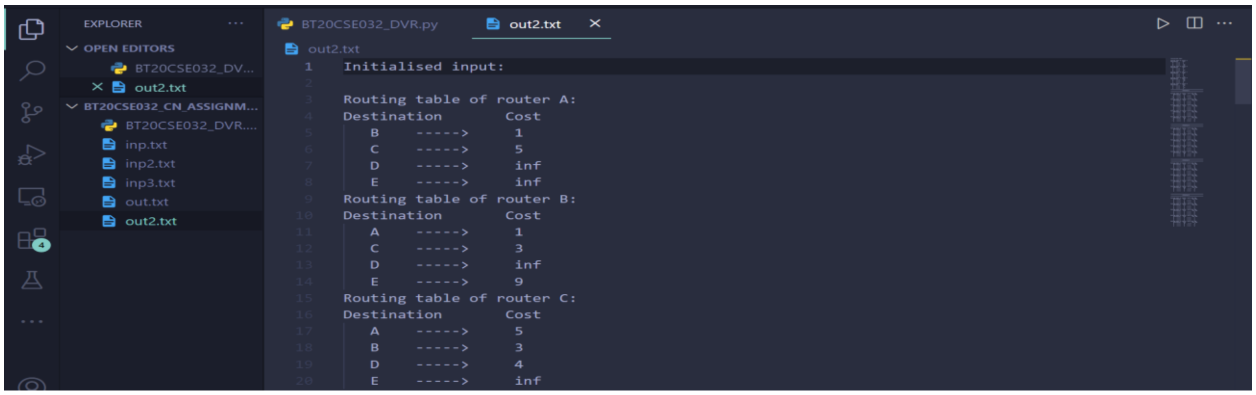


```
5
A B C D E
A B 1
A C 5
B C 3
C D 4
B E 9
D E 2
EOF
```

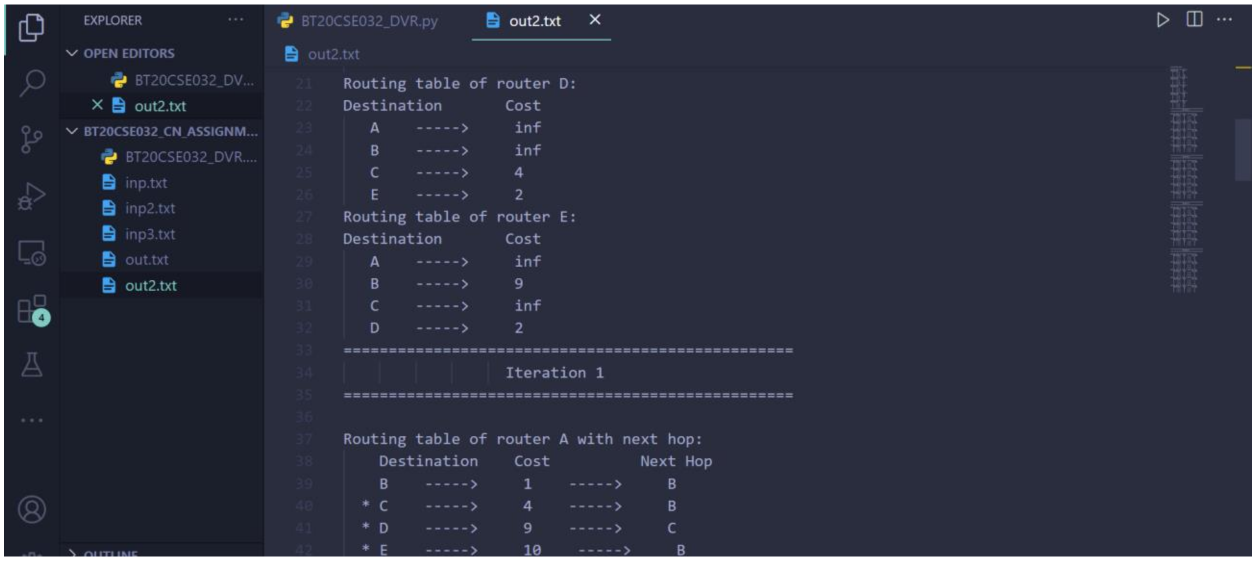
```
PS D:\BT20CSE032_CN_Assignment3> python BT20CSE032_DVR.py inp2.txt > out2.txt
PS D:\BT20CSE032_CN_Assignment3>
```



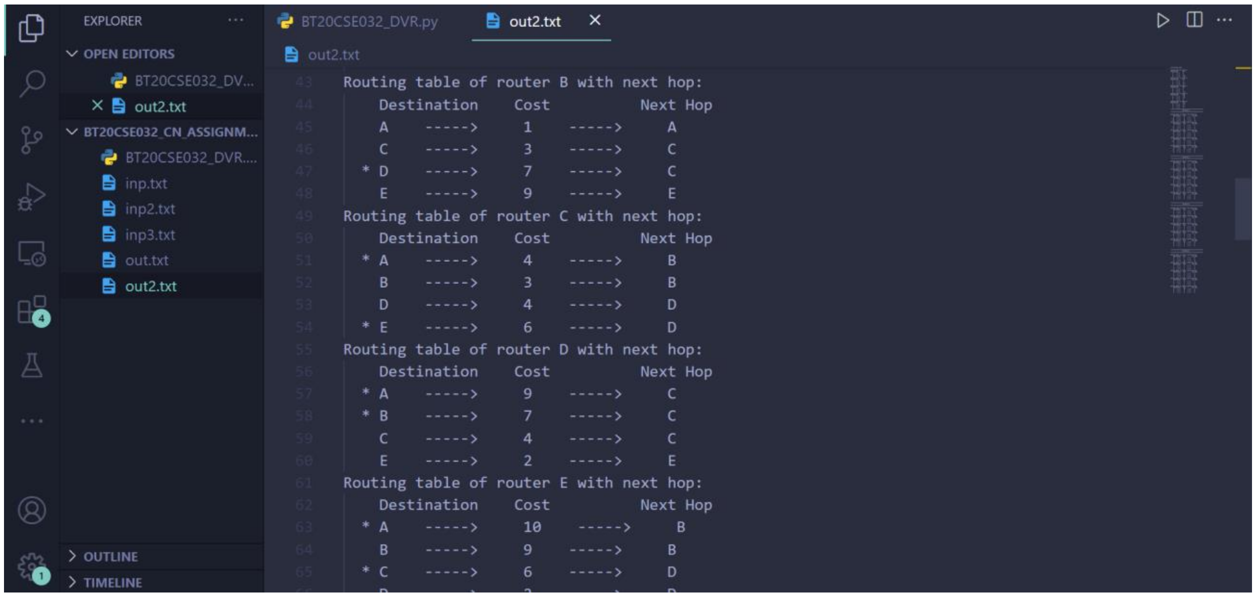
Output:



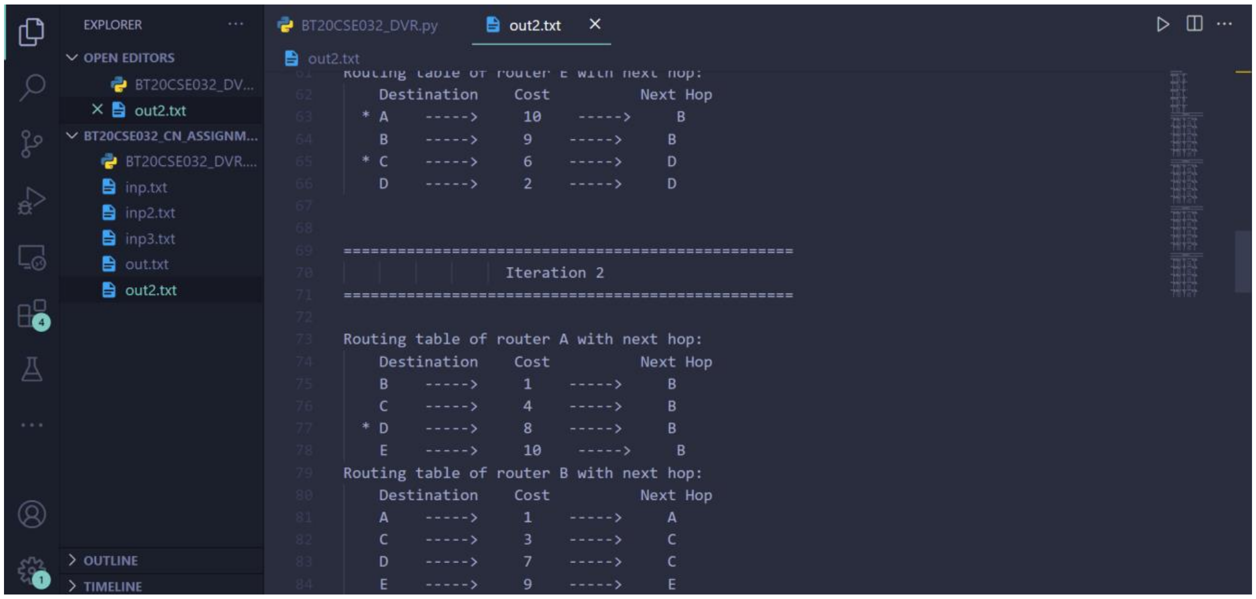
```
Initialised input:

Routing table of router A:
Destination       Cost
   B     ----->     1
   C     ----->     5
   D     ----->     inf
   E     ----->     inf
Routing table of router B:
Destination       Cost
   A     ----->     1
   C     ----->     3
   D     ----->     inf
   E     ----->     9
Routing table of router C:
Destination       Cost
   A     ----->     5
   B     ----->     3
   D     ----->     4
   E     ----->     inf
```



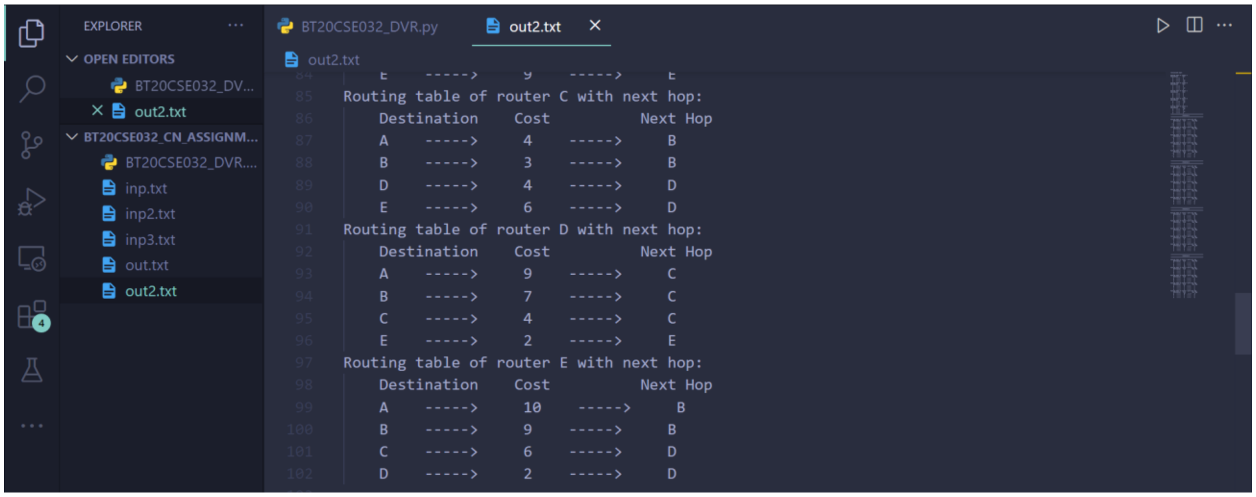
```
Routing table of router D:
Destination       Cost
   A     ----->     inf
   B     ----->     inf
   C     ----->     4
   E     ----->     2
Routing table of router E:
Destination       Cost
   A     ----->     inf
   B     ----->     9
   C     ----->     inf
   D     ----->     2
==================================================
                  Iteration 1
==================================================

Routing table of router A with next hop:
    Destination     Cost          Next Hop
    B     ----->     1     ----->     B
  * C     ----->     4     ----->     B
  * D     ----->     9     ----->     C
  * E     ----->     10     ----->     B
```



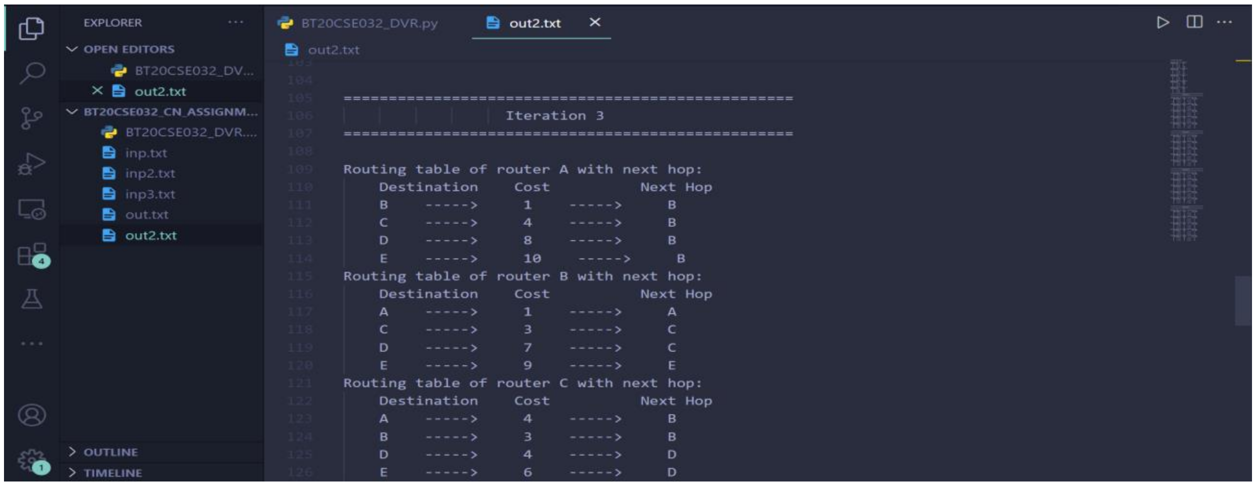
```
Routing table of router B with next hop:
    Destination    Cost         Next Hop
    A    ----->     1    ----->     A
    C    ----->     3    ----->     C
  * D    ----->     7    ----->     C
    E    ----->     9    ----->     E
Routing table of router C with next hop:
    Destination    Cost         Next Hop
  * A    ----->     4    ----->     B
    B    ----->     3    ----->     B
    D    ----->     4    ----->     D
  * E    ----->     6    ----->     D
Routing table of router D with next hop:
    Destination    Cost         Next Hop
  * A    ----->     9    ----->     C
  * B    ----->     7    ----->     C
    C    ----->     4    ----->     C
    E    ----->     2    ----->     E
Routing table of router E with next hop:
    Destination    Cost         Next Hop
  * A    ----->     10    ----->     B
    B    ----->     9    ----->     B
  * C    ----->     6    ----->     D
```



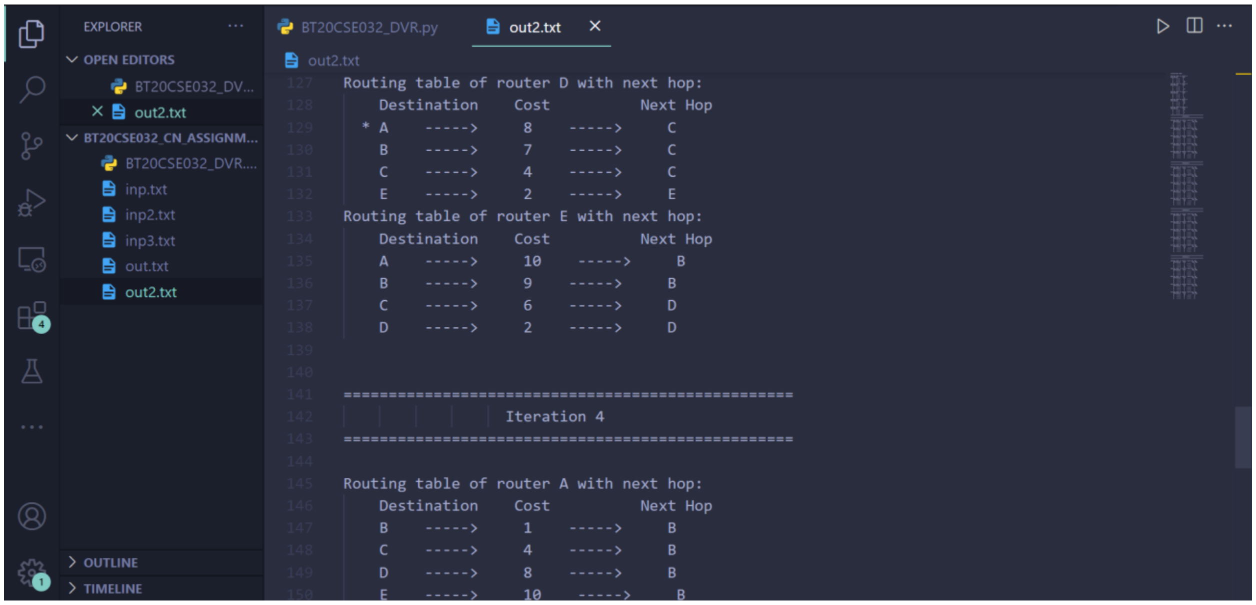
```
Routing table of router E with next hop:
    Destination    Cost         Next Hop
  * A    ----->     10    ----->     B
    B    ----->     9    ----->     B
  * C    ----->     6    ----->     D
    D    ----->     2    ----->     D


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

Routing table of router A with next hop:
    Destination    Cost         Next Hop
    B    ----->     1    ----->     B
    C    ----->     4    ----->     B
  * D    ----->     8    ----->     B
    E    ----->     10    ----->     B
Routing table of router B with next hop:
    Destination    Cost         Next Hop
    A    ----->     1    ----->     A
    C    ----->     3    ----->     C
    D    ----->     7    ----->     C
    E    ----->     9    ----->     E
```



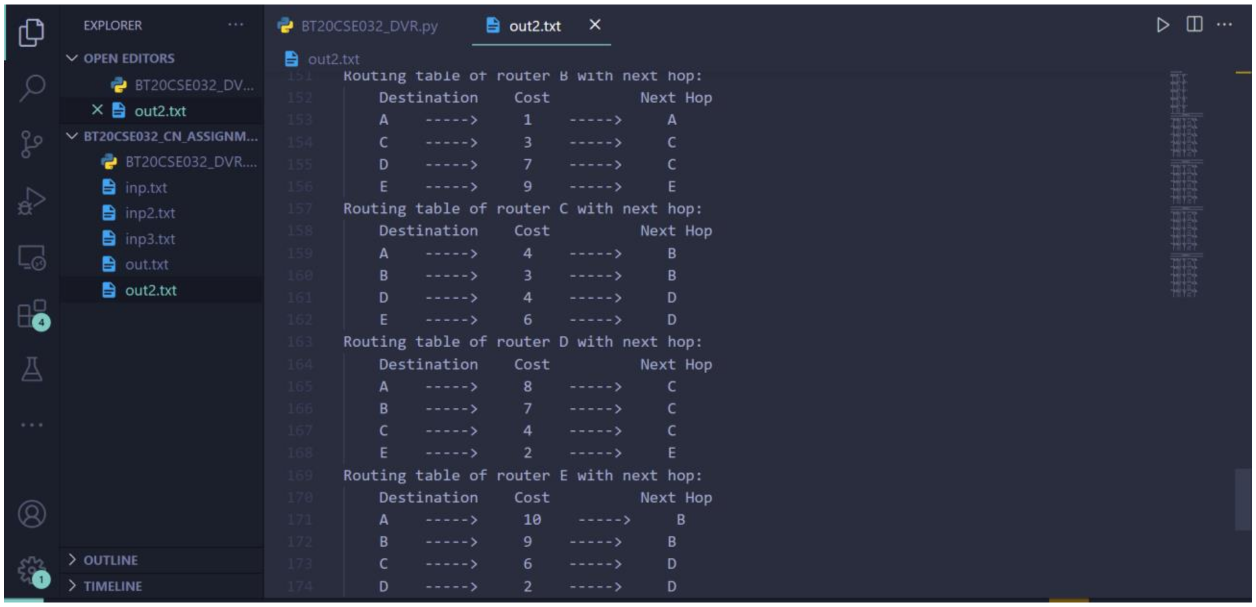
```
    E     ----->    9    ----->    E
Routing table of router C with next hop:
    Destination     Cost          Next Hop
    A     ----->    4    ----->    B
    B     ----->    3    ----->    B
    D     ----->    4    ----->    D
    E     ----->    6    ----->    D
Routing table of router D with next hop:
    Destination     Cost          Next Hop
    A     ----->    9    ----->    C
    B     ----->    7    ----->    C
    C     ----->    4    ----->    C
    E     ----->    2    ----->    E
Routing table of router E with next hop:
    Destination     Cost          Next Hop
    A     ----->    10    ----->    B
    B     ----->    9    ----->    B
    C     ----->    6    ----->    D
    D     ----->    2    ----->    D
```



```

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

Routing table of router A with next hop:
    Destination     Cost          Next Hop
    B     ----->    1    ----->    B
    C     ----->    4    ----->    B
    D     ----->    8    ----->    B
    E     ----->    10    ----->    B
Routing table of router B with next hop:
    Destination     Cost          Next Hop
    A     ----->    1    ----->    A
    C     ----->    3    ----->    C
    D     ----->    7    ----->    C
    E     ----->    9    ----->    E
Routing table of router C with next hop:
    Destination     Cost          Next Hop
    A     ----->    4    ----->    B
    B     ----->    3    ----->    B
    D     ----->    4    ----->    D
    E     ----->    6    ----->    D
```



```
Routing table of router D with next hop:
     Destination     Cost            Next Hop
   * A     ----->      8     ----->      C
     B     ----->      7     ----->      C
     C     ----->      4     ----->      C
     E     ----->      2     ----->      E
Routing table of router E with next hop:
     Destination     Cost            Next Hop
     A     ----->      10     ----->      B
     B     ----->      9     ----->      B
     C     ----->      6     ----->      D
     D     ----->      2     ----->      D


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

Routing table of router A with next hop:
     Destination     Cost            Next Hop
     B     ----->      1     ----->      B
     C     ----->      4     ----->      B
     D     ----->      8     ----->      B
     E     ----->      10     ----->      B
```



```
Routing table of router B with next hop:
     Destination     Cost            Next Hop
     A     ----->      1     ----->      A
     C     ----->      3     ----->      C
     D     ----->      7     ----->      C
     E     ----->      9     ----->      E
Routing table of router C with next hop:
     Destination     Cost            Next Hop
     A     ----->      4     ----->      B
     B     ----->      3     ----->      B
     D     ----->      4     ----->      D
     E     ----->      6     ----->      D
Routing table of router D with next hop:
     Destination     Cost            Next Hop
     A     ----->      8     ----->      C
     B     ----->      7     ----->      C
     C     ----->      4     ----->      C
     E     ----->      2     ----->      E
Routing table of router E with next hop:
     Destination     Cost            Next Hop
     A     ----->      10     ----->      B
     B     ----->      9     ----->      B
     C     ----->      6     ----->      D
     D     ----->      2     ----->      D
```