BT Interview Coding Exercise – Least Cost Routing

Develop a Java application to determine the optimal route in a network for establishing a call between two individuals. A route is considered the best route when it has the lowest cost in the network. Your solution should have a class called "LeastCostRouting" which exposes at least the following methods:

- **public static void main(String[] args)**
  - reads the CSV files containing the network data (Network Element, Links, Person, and Network) from local disk and maintain them using appropriate data structures. Format of CSV is described in appendix section.
  - takes name of “Source Person” and “Destination Person” as user input using System input stream.
  - finds appropriate source and destination Person Objects from the above data structure holding “Person” objects based on the names taken from user using System I/O.
- **private void findLeastCostRoute(Person sourcePerson, Person destinationPerson)**
  - is called by the main method by passing the source and destination Person Objects.
  - finds the route with least cost and prints the details.

Please define the necessary entities for this solution. You may add additional classes if needed to maintain a clean and standard design. Aim to utilize the Java Stream API and functional programming techniques wherever possible.

Explanation of the Problem:

- A typical network consists of Network Elements (NE) and Links/Cables that connect these Network Elements.
- A Network Element is essentially a physical server that processes a telephonic call.
- Each Network Element (NE) requires a certain amount of time to process a call, referred to as processing time (PT), measured in milliseconds (ms).
- Network Elements are connected to each other via Links, labeled as Link12, Link23, etc.
- The cost of using a link is specified in the diagram, with the price measured in pence (p).
- Note that the connection between the Exchange and a Network Element is not considered a Link and, therefore, has no associated cost. This connection is depicted as a dotted line in the diagram provided in the Appendix.
- Not all Network Elements are connected to the Exchange, which should be considered when finding or building routes.
- When a person makes a call, it utilizes the nearest exchange, followed by the Network Elements and Links in the network, to finally connect to the destination number. For simplicity, persons are placed near their respective exchanges in the provided diagram and the same information is available in the Person CSV file.
- Every person can make or receive a call.
- Define all necessary entities, such as Person, NetworkElement, Route, etc., and create the objects according to the scenario described in the Appendix.
- Use appropriate data structures to store Network Elements, Links, routes, etc.
- Both processing time and price are important in terms of cost: processing time is critical for the customer, while the price for using a Link is important for BT.So, the formula for calculating the Cost of a Route is as following.

  Cost of a Route = (5X Total Processing Time) + (2 X Total Price)

2 sample networks can be found in the Appendix.

Submitting Your Code

Email your full submission source code as a zipped attachment. You can also include with your program code any other data files, unit tests and so on to demonstrate that it works – but do not include any pre-compiled files (*.jar, *.class, etc.) as your submission may be blocked by our corporate firewalls - even if the files are contained within a zip file. Please mention all the steps required for running the code.

Other Requirements

Your program must not rely on any libraries that do not form part of a normal base installation of the language, but you may use anything from within the language's standard libraries. So, for example, unless explicitly asked to do so, we wouldn't expect you to implement your own sort routine if there is already one available in the standard libraries. Please be aware that your program should, where possible, avoid operating system-specific assumptions such line ending characters or file-name path delimiters.

We will test your code and expect you to have done so prior to submitting it. You will not automatically "fail" if we uncover bugs or limitations in your program. However, please be prepared to be challenged over improvements and possible fixes when we discuss your code with you. We will test your code with input sets other than those supplied in the problem description, so please expect that.

You may find that the specification of the problem seems ambiguous, or not as complete as you would like. That is, of course, part of the reality of software development! If you find that you have to make certain assumptions in order to implement your program, then please do so, but make it clear what those assumptions are. We will not provide you with further examples or test cases – although if you find something in the problem specification that seems contradictory or just plain wrong then please let us know.

It should go without saying that the program must be your own work. You are free to take inspiration and even small code snippets from other sources, but you must acknowledge if you do so with appropriate comments. In any event you will be expected to be able to explain in detail how every part of your program works.

Appendix - Sample Networks

The diagrams below represent 2 sample BT Networks. The program will be used for finding least cost route for one Network at a time.

- In the Network 1.1, The nearest exchanges for Person A and B are Exchange1 and Exchange2 respectively. The possible routes for making a call between Person A and B are:
    - NE1->NE7->NE8
    - NE1->NE7->NE2->NE3
    - NE1->NE2-NE3
    - NE7->NE2->NE3
    - NE7->NE8
    - NE7->NE1->NE2->NE3
- Route having Least cost for making a call is (NE7->NE8) and the cost for the same is calculated as below:

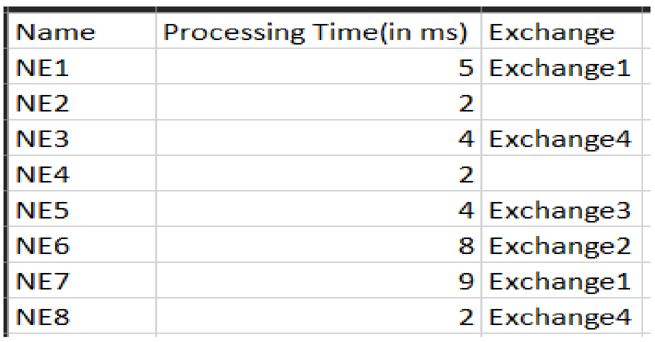
  Cost =5X (9+2) + 2X (2) = 59 units.
- In network 1.2, the route having least cost for making a call between Person B and Person D is (NE6->NE4->NE3) and the cost for the same is calculated as below:

  Cost=5X (8+2+4) +2X (4+2) =82 units
- The input data for your test cases should be taken from the below networks.
- The network is dynamic in nature and provides test data for the test cases. The logic of the program remains same.
- The network data needs to be taken from 4 CSV files which are stored in the local disk:
    - Network Element CSV file which has the details of network elements
    - Links CSV File which has the details of Links
    - Person CSV file which has the details of Person

Coding Exercise – Least Cost Routing version1.2

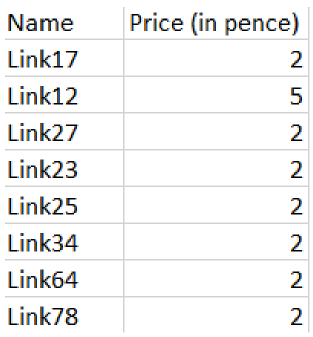
    - Network CSV file which has the details of Network

- The below CSV files represent the data for the Network 1.1.
- You can create similar files for Network1.2.



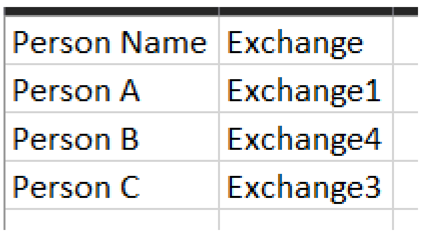
Name	Processing Time(in ms)	Exchange
NE1	5	Exchange1
NE2	2	
NE3	4	Exchange4
NE4	2	
NE5	4	Exchange3
NE6	8	Exchange2
NE7	9	Exchange1
NE8	2	Exchange4

Network Element CSV File 1



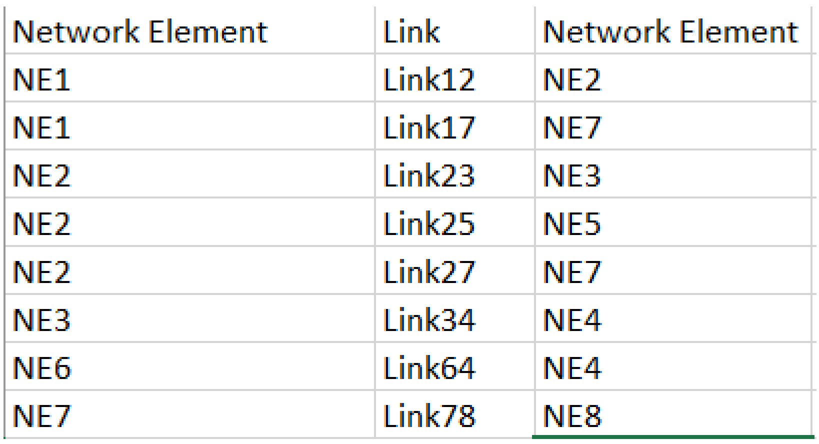
Name	Price (in pence)
Link17	2
Link12	5
Link27	2
Link23	2
Link25	2
Link34	2
Link64	2
Link78	2

Links CSV File 1



Person Name	Exchange
Person A	Exchange1
Person B	Exchange4
Person C	Exchange3

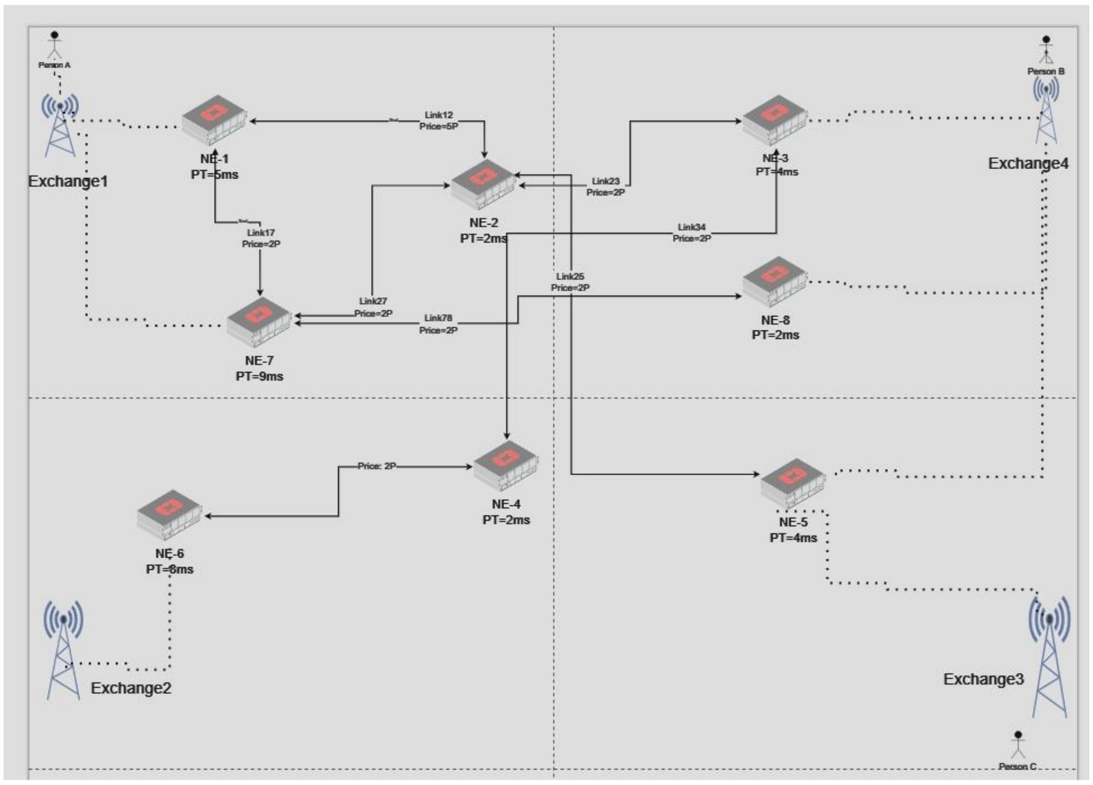
Person CSV File 1



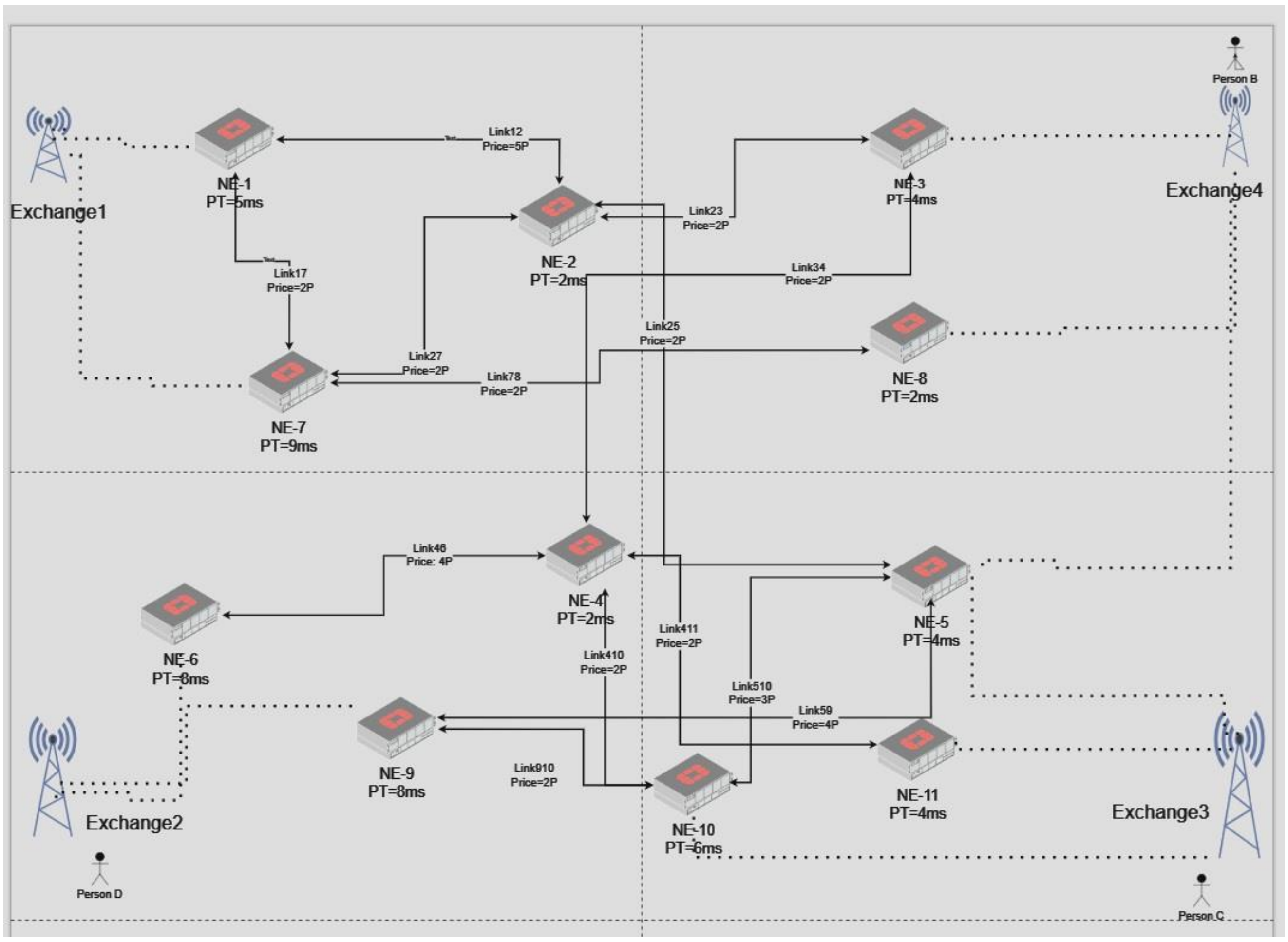
Network Element	Link	Network Element
NE1	Link12	NE2
NE1	Link17	NE7
NE2	Link23	NE3
NE2	Link25	NE5
NE2	Link27	NE7
NE3	Link34	NE4
NE6	Link64	NE4
NE7	Link78	NE8

Network CSV file 1

Coding Exercise – Least Cost Routing version1.2



Network 1 1



Network 1 2