

Networks : social networks, metro plan, etc...

The goal of the project is to play with the notion of networks. It is something that we interact with all the times : people participates in social networks, take the metro, academics may also be interested in citation networks, etc.

Networks is also something that is studied in computer science and mathematics in the field of graph theory. The goal of the project is to make you code in java.

As a hint, consider the idea of the `LinkedList` presented in the slides about generic types and collection. In a linked list, each element has only one next element. In graph, it may just have more elements.

You will be provided with two input files. One will represent the network of the parisian métro. Another one will be a model of a social network (let's imagine it is `linkedIn`). The goal of the project is that you write code for answering the following questions :

1. Write the first 10 stations in decreasing number of the number of connections.
2. Write a method that finds an itinerary to go from one station to another and print on the console all the stations on the way (ex : if you go from "Porte Dauphine" to "Trocadéro", you should write "Porte Dauphine", "Victor Hugo", "Charles de Gaulle – Etoile", "Kléber", Boissière", "Trocadero").
3. Using the work you have done in the previous questions, implement a code that can
 - (a) write the names of the 10 persons that are the most connected, in descending order.
 - (b) write the encounters that are necessary to introduce a first person to a second. Do you observe the famous six degrees of separation (or six degrees of Kevin Bacon) ?

For question 2, your solution does not have to be optimal. By optimality, we mean that there is no other itinerary that would go through less stations. of course, it is better if it is, but it is not required. You will get bonus points if the itinerary are optimal.

The files provides will have the following structure :

- a line starting with the character '%' is a comment
- the first part of the files provides the name of each station
- the second part of the file provides each edge of the metro
- the two parts are separated with a line starting with the characters .
- for a station, we only have its name
- for an edge, the names of the two stations are separated with a semi column. For instance
Porte Dauphine:Victor Hugo
Victor Hugo:Charles de Gaulles - Etoile

The same structure will be used for the social network (except that station's names will be replaced by name of people).

Note that you could find useful the method `split` in the class `String`.

We provide the code to read the input files. The input files should be saved in the same directory where your project is.