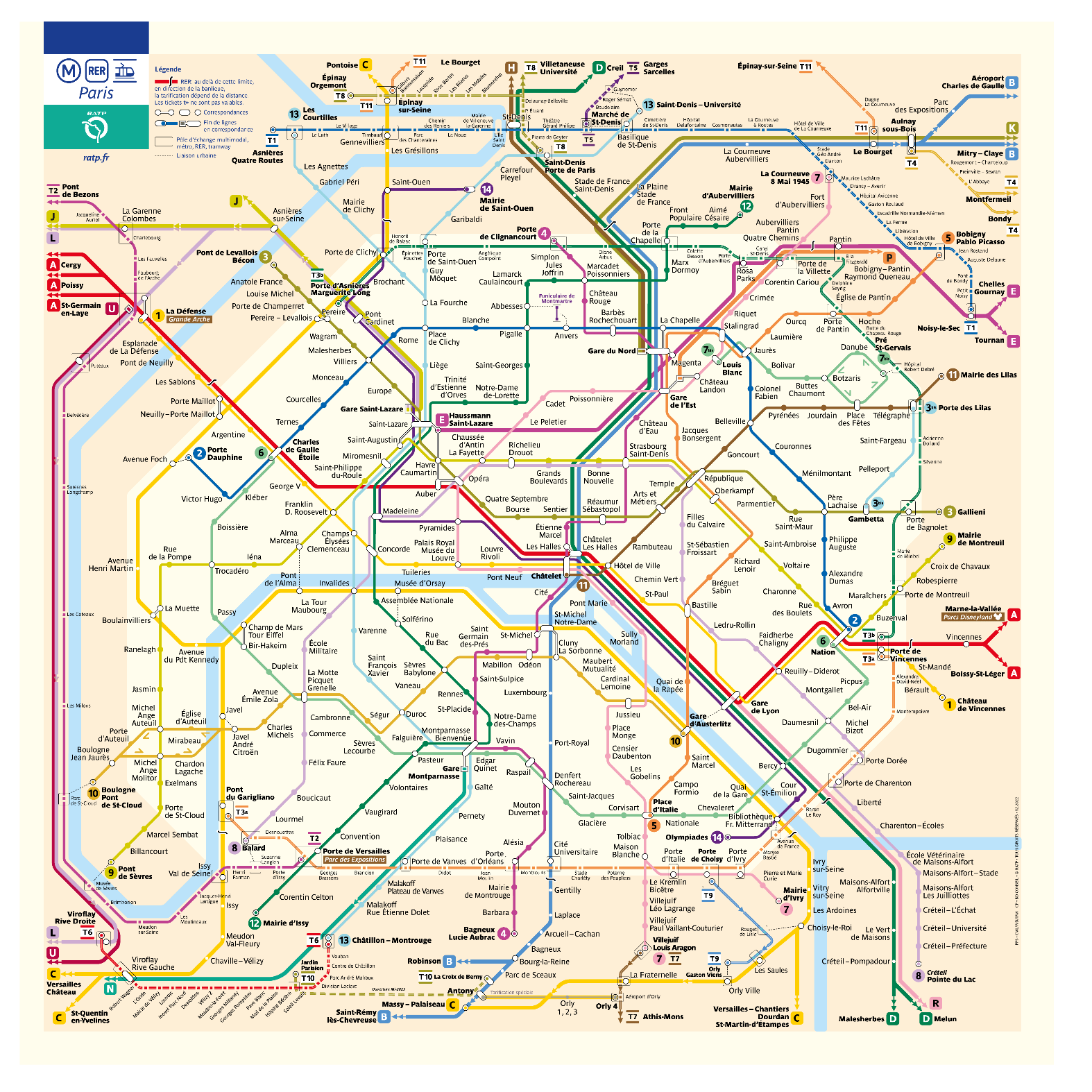
**JOURNEY PLANNER FOR PARIS METRO**

****

Paris Metro is the [second busiest metro system](https://en.wikipedia.org/wiki/List_of_metro_systems) in [Europe](https://en.wikipedia.org/wiki/Europe), after the [Moscow Metro](https://en.wikipedia.org/wiki/Moscow_Metro), as well as the tenth-busiest in the world [1]. It carried 1.498 billion passengers in 2019, roughly 4.1 million passengers a day, which makes it the most used public transport system in Paris [2].  As of the end of May 2022, there are a total of 308 stations on 16 different lines [3].

A journey planner (or trip planner) is a specialized electronic search engine that finds one or more journey (trip) suggestions between an origin and a destination. This system assists travelers in planning their journey.

1. **Journey Planner Search Engine**

In this assignment, you are expected to develop an algorithm to find the best journey between the given ***origin*** and ***destination*** stops. You should consider two optimization criteria: ***fewer stops*** and ***minimum time***. In the first case, you should use equal edge weights in the graph. In the second case, you should use the given time intervals in the consecutive stops. You should represent each metro station as a node and represent each line connecting two consecutive stops in a certain direction as a directed edge to form a transportation graph. This graph is a directed graph as illustrated in Figure 1.

**A diagram of a person walking towards a couple of circles

Description automatically generated**

Figure 1. Illustration of a transportation graph.

To make point to point queries in a transportation network, some sort of walk(transfer) edges are required, so any stage of the journey can be covered by walk or passengers may walk between the stops while transferring between two different lines. Walk-distance edges are also providing to link each of the transportation networks (bus, train, metro, ferry etc.). Two stops u and v are labeled as neighbor stops by adding walk-distance edges between them, if a road segment is available to pedestrians and s less than the maximum allowed walking distance.

A sample query from the origin station “Charles de Gaulle” to destination station “Odéon” returns following result.

|  |
| --- |
| Origin station: Charles de Gaulle  Destination station: Odéon  Preferetion: Minimum Time  Suggestion  Line 1:  Charles de Gaulle-Etoile – Châtelet (8 stations)  Line 4  Châtelet – Odéon (3 stations)  21 min |

1. **Bonus**
2. **Limiting the transfers**

You should modify your algorithm to limit transfers. Thus, the alternative paths include only direct routes and the routes containing up to two transfers. Two consecutive walks are not allowed.

1. **Suggesting more than one alternative**

You should modify your algorithm to suggest more than one alternative (best paths up to five). Several runs of the algorithm are required to find alternative paths.

1. **Provided Resources**

* Paris\_RER\_Metro.csv (Contains all metro lines and A and B RER lines)

1. **Requirements**

* Usage of *Java* programming language and *Edge List Graph Data Structure* are required.
* *Object Oriented Programming (OOP)* principles must be applied.
* *Exception handling* must be used when it is needed.

**References**

1. <https://www.ratp.fr/plan-metro>
2. <https://en.wikipedia.org/wiki/Paris_M%C3%A9tro>
3. <https://en.wikipedia.org/wiki/List_of_Paris_M%C3%A9tro_stations>
4. <https://prim.iledefrance-mobilites.fr/en/catalogue-data>