

Projet Structures de données : avancé

1 Objectif

Les fichiers `lignes.txt` et `troncons.txt` contiennent des données à propos du réseau de la STIB (Société des Transports Intercommunaux de Bruxelles).

Ces données forment un graphe dirigé. Les sommets sont des stations (ex : ALMA). Il y a un arc entre deux stations s'il existe un tronçon reliant deux stations.

L'objectif général du projet est d'implémenter un programme java permettant de calculer des itinéraires entre deux stations.

Voici un exemple d'itinéraire :

```
Deplacement [ligne=25, depart=BOILEAU, arrivee=MONTGOMERY, attenteMoyenne=7, duree=1,
nbTroncon=1, type=tram, direction=ROGIER]
Deplacement [ligne=1, depart=MONTGOMERY, arrivee=ALMA, attenteMoyenne=8, duree=9,
nbTroncon=6, type=metro, direction=STOCKEL]
nbTroncons=7
dureeTransport=10 dureeTotale=25
```

Ceci représente un trajet de la station BOILEAU à la station ALMA. Il est constitué de deux déplacements : un déplacement avec le tram 25 (direction ROGIER) jusqu'à la station MONTGOMERY et un deuxième avec le métro 1 (direction STOCKEL) jusqu'à la station ALMA. La durée du trajet est de 25 minutes dont 10 minutes dans les transports. Ce trajet est constitué de 7 tronçons.

2 Sur Moodle

Pour mener à bien votre projet, nous vous fournissons plusieurs fichiers :

- `lignes.txt` contient des informations à propos de différentes lignes de la STIB. Il contient une ligne par ligne. Pour chaque ligne, les informations suivantes sont séparées par des virgules dans cet ordre : un identifiant unique, le numéro, la première station, la destination, le type de transport (métro, tram ou bus) et le temps d'attente moyen.
- `troncons.txt` contient des informations à propos des tronçons. Il contient une ligne par tronçon. Pour chaque tronçon, les informations suivantes sont séparées par des virgules dans cet ordre : le numéro de la ligne, le départ du tronçon, l'arrivée du tronçon et la durée du tronçon
- une classe `Main.java` que vous ne pouvez pas modifier. Le code de cette classe est présenté ci-dessous. La classe à construire `Graph` devra avoir un constructeur qui prendra les deux fichiers contenant les informations des lignes et des tronçons en paramètre. Elle devra également contenir deux méthodes pour calculer un itinéraire. Ces deux méthodes prennent deux paramètres : les noms des stations de départ et d'arrivée.

```
import java.io.File;

public class Main {
    public static void main(String[] args) {
```

```

    try {
        File lignes = new File("lignes.txt");
        File troncons = new File("troncons.txt");
        Graph g = new Graph(lignes, troncons);
        g.calculerCheminMinimisantNombreTroncons("BOILEAU", "ALMA");
        System.out.println("-----");
        g.calculerCheminMinimisantTempsTransport("BOILEAU", "ALMA");
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

3 Tâches à effectuer

Nous vous demandons de rendre la classe Main fonctionnelle en réalisant les différentes tâches suivantes :

- Implémenter la méthode `calculerCheminMinimisantNombreTroncons` qui calcule le chemin entre deux stations avec le moins de tronçons possibles (voir point 4). S'il est impossible d'aller d'une station à une autre, votre programme lancera une exception.
- Implémenter la méthode `calculerCheminMinimisantTempsTransport` qui calcule le chemin entre deux stations avec le moins de temps passé dans les transports (voir point 5). S'il est impossible d'aller d'une station à une autre, votre programme lancera une exception.

4 Itinéraire avec le moins de tronçons

La méthode `calculerCheminMinimisantNombreTroncons` calculera et affichera à la console les itinéraires qui minimise le nombre de tronçons.

Par exemple, pour aller de la station BOILEAU à la station ALMA, l'itinéraire le plus court contient 7 tronçons.

```

Troncon [depart=BOILEAU, arrivee=MONTGOMERY, duree=1, ligne=Ligne [id=68, nom=25,
source=BOONDAEL GARE, destination=ROGIER, type=tram, attenteMoyenne=7]]
Troncon [depart=MONTGOMERY, arrivee=JOSEPH.-CHARLOTTE, duree=2, ligne=Ligne [id=427, nom=1,
source=GARE DE L'OUEST, destination=STOCKEL, type=metro, attenteMoyenne=8]]
Troncon [depart=JOSEPH.-CHARLOTTE, arrivee=GRIBAUMONT, duree=1, ligne=Ligne [id=427, nom=1,
source=GARE DE L'OUEST, destination=STOCKEL, type=metro, attenteMoyenne=8]]
Troncon [depart=GRIBAUMONT, arrivee=TOMBERG, duree=1, ligne=Ligne [id=427, nom=1,
source=GARE DE L'OUEST, destination=STOCKEL, type=metro, attenteMoyenne=8]]
Troncon [depart=TOMBERG, arrivee=ROODEBEEK, duree=2, ligne=Ligne [id=427, nom=1,
source=GARE DE L'OUEST, destination=STOCKEL, type=metro, attenteMoyenne=8]]
Troncon [depart=ROODEBEEK, arrivee=VANDERVELDE, duree=2, ligne=Ligne [id=427, nom=1,
source=GARE DE L'OUEST, destination=STOCKEL, type=metro, attenteMoyenne=8]]
Troncon [depart=VANDERVELDE, arrivee=ALMA, duree=2, ligne=Ligne [id=327, nom=N05,
source=GARE CENTRALE, destination=KRAAINEM, type=bus, attenteMoyenne=22]]
nbTroncons=7
dureeTransport=11 dureeTotale=48

```

Il est peut-être possible de trouver d'autres itinéraires avec 7 tronçons.

5 Itinéraire avec le moins de temps passé dans les transports

La méthode `calculerCheminMinimisantTempsTransport` calculera et affichera à la console les itinéraires qui minimise le temps passé dans les transports.

Pour aller de BOILEAU à ALMA, il est possible de trouver un itinéraire avec 10 minute passé dans les transports.

```
Troncon [depart=BOILEAU, arrivee=MONTGOMERY, duree=1, ligne=Ligne [id=68, nom=25,
source=BOONDAEL GARE, destination=ROGIER, type=tram, attenteMoyenne=7]]
Troncon [depart=MONTGOMERY, arrivee=JOSEPH.-CHARLOTTE, duree=2, ligne=Ligne [id=427, nom=1,
source=GARE DE L'OUEST, destination=STOCKEL, type=metro, attenteMoyenne=8]]
Troncon [depart=JOSEPH.-CHARLOTTE, arrivee=GRIBAUMONT, duree=1, ligne=Ligne [id=427, nom=1,
source=GARE DE L'OUEST, destination=STOCKEL, type=metro, attenteMoyenne=8]]
Troncon [depart=GRIBAUMONT, arrivee=TOMBERG, duree=1, ligne=Ligne [id=427, nom=1,
source=GARE DE L'OUEST, destination=STOCKEL, type=metro, attenteMoyenne=8]]
Troncon [depart=TOMBERG, arrivee=ROODEBEEK, duree=2, ligne=Ligne [id=427, nom=1,
source=GARE DE L'OUEST, destination=STOCKEL, type=metro, attenteMoyenne=8]]
Troncon [depart=ROODEBEEK, arrivee=VANDERVELDE, duree=2, ligne=Ligne [id=427, nom=1,
source=GARE DE L'OUEST, destination=STOCKEL, type=metro, attenteMoyenne=8]]
Troncon [depart=VANDERVELDE, arrivee=ALMA, duree=1, ligne=Ligne [id=427, nom=1, source=GARE
DE L'OUEST, destination=STOCKEL, type=metro, attenteMoyenne=8]]
nbTroncons=7
dureeTransport=10 dureeTotale=25
```

6 Tâches bonus

Vous pouvez proposer une meilleure sortie pour l'utilisateur en fusionnant tous les tronçons appartenant à la même ligne dans un seul déplacement. Voici un exemple de sortie pour l'itinéraire avec le moins de tronçons :

```
Deplacement [ligne=25, depart=BOILEAU, arrivee=MONTGOMERY, attenteMoyenne=7, duree=1,
nbTroncon=1, type=tram, direction=ROGIER]
Deplacement [ligne=1, depart=MONTGOMERY, arrivee=VANDERVELDE, attenteMoyenne=8, duree=8,
nbTroncon=5, type=metro, direction=STOCKEL]
Deplacement [ligne=N05, depart=VANDERVELDE, arrivee=ALMA, attenteMoyenne=22, duree=2,
nbTroncon=1, type=bus, direction=KRAAINEM]
nbTroncons=7
dureeTransport=11 dureeTotale=48
```

7 Organisation et livrables

Ce projet se déroule du 6 mars 2022 au 26 mars 2022 par groupe de deux étudiants. Les étudiants d'un même groupe doivent être dans la même série. Durant tout le projet, la présence aux cours est obligatoire.

Si une série comporte un nombre impair d'étudiant, nous accepterons un seul groupe de 3 étudiants.

Le projet est à remettre via « Moodle » pour le dimanche 26 mars 2022 à 12h00 (midi). Nous ne demandons pas de rapport. Si certaines choses méritent des explications, vous pouvez les fournir en commentaire dans les différents fichiers.

Le plagiat sera lourdement sanctionné (au minimum 0 à l'UE). Tous les projets seront automatiquement testés par un logiciel de détection de plagiat.