

Rapport Projet Informatique du Métro

Bouarah Romain Langdorph Matthieu Ketels Lucas
Souffan Nathan

April 30, 2020

Dans le cadre du Projet Informatique du 4ème semestre, voici le projet Métro. Ce programme permet de donner le trajet le plus court lors d'un déplacement sur un réseau de métro, représenté par un graphe. Ce programme a été codé en Java.

Contents

I	Guide d'utilisation du programme	3
1	Installation	3
2	Utilisation	3
3	Exemple d'exécution	4
II	Projet Métro	8
4	Présentation générale	8
4.1	Modélisation	8
4.2	Organisation	8
4.3	Problèmes rencontrés	8
5	Le projet dans ses grandes lignes	8
5.1	Les différents algorithmes	8
5.2	Structure des classes	8

Part I

Guide d'utilisation du programme

Voici des indications pour utiliser notre programme "metro". Nous l'avons conçu avec Maven, JUnit et Java.

1 Installation

Avant d'installer le programme, il faut vérifier bien avoir Java JDK, Maven et JUnit d'installer sur son ordinateur. Sinon les installer. Il faut également vérifier que les versions sont suffisamment à jour, version au moins 8.0 pour Java, au moins 3.6 pour Maven, et au moins 4.11 pour JUnit. Vous pouvez alors cloner le projet depuis master dans les fichiers de votre ordinateur. Vous êtes alors prêt pour utiliser "metro".

2 Utilisation

Une fois la partie installation effectuée, vous pouvez compiler et exécuter "metro". Se placer dans le répertoire adapté et se positionner sur master(par défaut). Pour compiler le projet dans son intégralité, il faut utiliser la commande "make". Il y a la possibilité de compiler juste datamodel ou le webserver, en faisant suivre leur nom après "make". On peut compiler les tests avec "make test". "make clean" supprime tous les fichiers produits pendant la compilation.

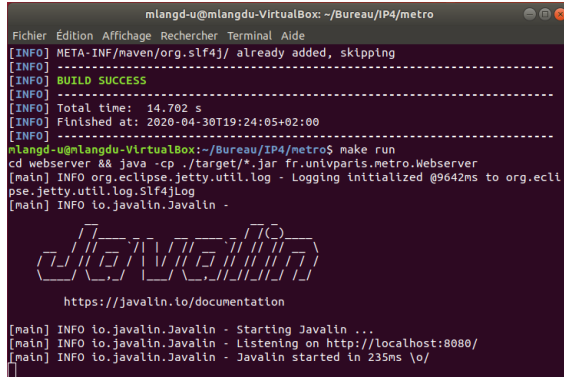
Pour exécuter le programme : on peut soit avoir une version minimale sur le terminal avec la commande "make run_terminal". Sinon on peut avoir la version site Web en utilisant la commande "make run".

Nous allons donc voir comment utiliser le webserver "metro" dans la section suivante.

Nous avons également l'option d'afficher le graphe du réseau d'une ville en utilisant la commande "make export_to_dot ARGS="NomDeLaVille" ". Notons que les réseaux de métro des villes de Lille, Lyon, Marseille, Rennes, Toulouse, Paris, et des aéroports de Charles de gaulle et Orly.

3 Exemple d'exécution

Nous compilons donc avec la commande "make". Nous lançons ensuite l'exécution avec "make run".

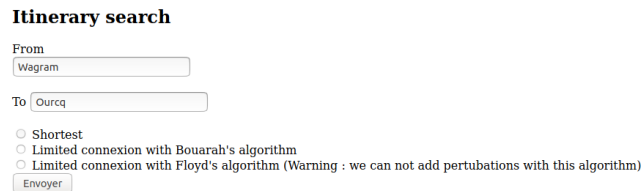


```
mclangd-u@mclangdu-VirtualBox: ~/Bureau/IP4/metro
[INFO] META-INF/maven/org.slf4j/ already added, skipping
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 14.702 s
[INFO] Finished at: 2020-04-30T19:24:05+02:00
[INFO] -----
mclangd-u@mclangdu-VirtualBox:~/Bureau/IP4/metro$ make run
cd webserver && java -cp ./target/*.jar fr.univparis.metro.Webserver
[main] INFO org.eclipse.jetty.util.log - Logging initialized @9642ms to org.ecl
pse.jetty.util.log.Slf4jLog
[main] INFO io.javalin.Javalin -
Javalin
https://javalin.io/documentation
[main] INFO io.javalin.Javalin - Starting Javalin ...
[main] INFO io.javalin.Javalin - Listening on http://localhost:8080/
[main] INFO io.javalin.Javalin - Javalin started in 235ms \o/
```

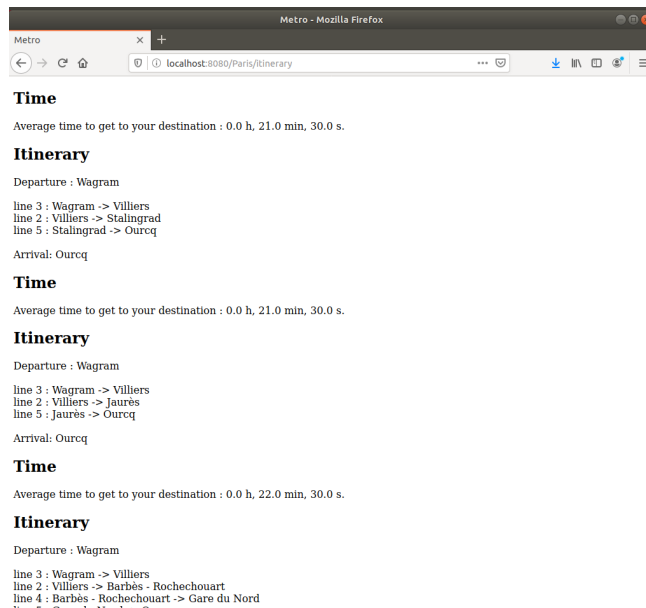
Il faut alors copier le lien(2ème ligne partant du bas) dans un navigateur web pour accéder au webserver de "metro". Cette page ci-dessous s'affiche et on sélectionne alors la ville de notre choix parmi celles possibles. On prend Paris dans cet exemple.



Nous pouvons alors rechercher un itinéraire. Ici de Wagram à Ourcq. Trois options s'offrent à nous : utiliser Dijkstra(Shortest) pour trouver le plus court chemin, l'algorithme de Floyd ou l'algorithme de Bouarrah pour trouver le plus court chemin avec un plafond de nombre de correspondances.



Nous faisons ici une recherche avec Shortest. S'affichent alors plusieurs itinéraires dans l'ordre croissant en fonction des temps de trajets.



Retournons à la page d'avant, on remarque la possibilité d'ajouter des perturbations de trafics. On peut arrêter ou ralentir une ligne ou une partie d'une ligne, fermer une station, fermer une ligne dans une station. Notre itinéraire précédent nous faisais passer par la ligne 2. Fermons donc la ligne 2 puis effectuons la même recherche de trajet entre Wagram et Ourcq.

Trafics Perturbation

Current Perturbation

Add perturbation

Line shutdown

Name of the perturbation

Line you want to shutdown

Line slowdown

Name of the perturbation

Line you want to slow down

Times which multiply every trajet in the line

Station shutdown

Name of the perturbation

Itinerary search

From

To

☒ Shortest
☐ Limited connexion with Bouarrah's algorithm
☐ Limited connexion with Floyd's algorithm (Warning : we can not add perturbations with this algorithm)

Trafics Perturbation

Current Perturbation

Here are the actual perturbation, select the one you want to remove

☐ Ligne 2 fermée

On peut à présent voir le trajet le plus court sachant que la ligne 2 est fermée.

Time

Average time to get to your destination : 0.0 h, 25.0 min, 30.0 s.

Itinerary

Departure : Wagram

line 3 : Wagram -> Havre - Caumartin
line 9 : Havre - Caumartin -> Chaussée d'Antin - La Fayette
line 7 : Chaussée d'Antin - La Fayette -> Gare de l'Est
line 5 : Gare de l'Est -> Ourcq

Arrival: Ourcq

Time

Average time to get to your destination : 0.0 h, 26.0 min, 0.0 s.

Itinerary

Departure : Wagram

line 3 : Wagram -> Opéra
line 7 : Opéra -> Gare de l'Est
line 5 : Gare de l'Est -> Ourcq

Arrival: Ourcq

Retournons maintenant à la page d'avant, allons en bas de page, et on peut voir qu'une section "Statistics" propose un lien vers une page affichant les statistiques du réseau de métro de la ville sélectionnée au préalable.

Statistics

To accede to the statistics of the network, click [here](#) and please wait few seconds.

Voici l'affichage de cette page :

Statistics

The longest trajet between two stations :

Traject : Pont de Sèvres Meta Station Start to Créteil - Préfecture Meta Station End
Time : 0.0h 54.0min 30.0sec

Number minimum of correspondence to do all the possible trajects on the network : 3

The average number of stations per line : 24

The line with the most stations : line 7

The line with the least stations : line 3bis

The average time from one terminus of a line to the other : 0.0h 33.0min 13.0sec

The longest(duration) line : line 8 in 0.0h 54.0min 0.0sec

The shortest(duration) line : line 3bis in 0.0h 4.0min 30.0sec

Voici la fin de cet exemple d'utilisation de "metro" via son webserver.

Part II

Projet Métro

4 Présentation générale

4.1 Modélisation

4.2 Organisation

4.3 Problèmes rencontrés

5 Le projet dans ses grandes lignes

5.1 Les différents algorithmes

5.2 Structure des classes

Diagramme des classes