

Université de Versailles - Saint Quentin
“Algorithmique de Graphes”
Le projet 2022 : *Métro, Boulot, Dodo*

1 Cadre

Ce projet est à réaliser en groupes de 2 ou 3 étudiants, voire exceptionnellement en solo. Il consiste à écrire un programme informatique permettant de déterminer le plus court itinéraire pour aller d’une station à une autre dans le métro Parisien. Ce programme peut être écrit dans le langage informatique de votre choix : C, C++, Java, Python, ...

A la fin du projet (avant les partiels du second semestre), il conviendra :

- de faire une démonstration (d’une dizaine de minutes environ) de votre programme devant l’enseignant de votre groupe de TD (si votre groupe de projet est composé d’étudiants de différents groupes de TD, vous pouvez choisir à quel enseignant vous avez le plus envie de faire cette magnifique démonstration).
- de lui rendre (ou de lui envoyer par mail) un rapport au **format pdf** décrivant le travail effectué, les structures de données choisies, les principales procédures, ... Le code du programme sera impérativement mis en annexe de ce rapport et il ne doit donc y avoir **qu’un seul fichier** envoyé. Il convient dans ce rapport de bien expliquer ce que vous avez fait, l’enseignant doit comprendre votre travail sans avoir à se plonger dans le détail de vos lignes de code.

2 Travail à réaliser

2.1 Les données

- La plupart des données sont à récupérer dans le fichier “metro.txt”.
- Ouvrir ce fichier, le regarder en détail, comprendre comment il est organisé. Pourquoi les sommets 7 et 8 sont-ils tous deux “Arts et Métiers” ?
- Ce fichier date et certaines nouvelles stations sur des lignes qui ont été rallongées n’y figurent pas. Il ne paraît pas nécessaire de les rajouter.
- Par contre, ce fichier est incomplet et il vous faudra rajouter des données qui n’y figurent pas comme les numéros de ligne de métro et les terminus des lignes.

2.2 Vérification de la connexité

Il est possible que le fichier metro.txt soit incomplet et que quelques rares liaisons y aient été omises. Votre premier travail de programmation doit donc consister à développer un module qui vérifie que votre graphe est bien connexe, c’est à dire qu’il est possible d’aller de n’importe quelle station à n’importe quelle autre. S’il s’avérait que des liaisons manquaient pour assurer cette connexité, à vous bien sur de les rajouter.

2.3 Le plus court itinéraire

C’est là le principal objectif de votre travail. Il convient de permettre à un utilisateur d’indiquer une station de départ et une station d’arrivée (remarque : ce doit être des stations et pas des stations sur une ligne) et

que votre programme calcule et indique le meilleur itinéraire pour aller de l'une à l'autre.

Résultat minimal :

L'utilisateur doit pouvoir saisir une station de départ et une station destination (dans l'exemple ci-dessous Arts et Métiers et Porte de Vincennes). L'itinéraire le plus court est alors calculé et son détail s'affiche, quelque chose comme :

- Vous êtes à Arts et Métiers.
- Prenez la ligne 3 direction Gallieni.
- A République, changez et prenez la ligne 8 direction Créteil.
- A Bastille, changez et prenez la ligne 1, direction Chateau de Vincennes.
- Vous devriez arriver à Porte de Vincennes dans 42 minutes.

Résultat avancé : une jolie interface

La carte du métro Parisien est affichée à l'écran (ou tout du moins une partie de celle-ci). L'utilisateur clique sur les stations de départ et d'arrivée et le plus court chemin s'affiche alors directement sur le plan de métro à l'écran.