

## Projet - Calculateur d'itinéraire pour le métro A rendre avant le Samedi 11 Février minuit



### 1 Enoncé du projet

Ce projet consistera à coder un algorithme permettant, étant donnée une station de métro de départ et une station de métro d'arrivée, de trouver le plus court chemin reliant les deux stations. Il vous est fourni deux fichiers de données avec le projet, que votre programme devra lire afin de construire la structure de données permettant de réaliser le calcul demandé.

### 2 Les fichiers de données

Les deux fichiers de données sont des fichiers CSV (comma separated values), où chaque ligne représente des données dont les champs sont séparés par des virgules. La première ligne est une ligne d'entête, qui peut être ignorée par votre programme.

Le fichier des stations contient, pour chaque station, son nom ainsi qu'un numéro d'identité unique. Le fichier des arêtes représente les "connexions directes" entre les stations. Chaque arête est caractérisée par un numéro de station d'origine, un numéro de station de destination, et le "numéro" de ligne auquel appartient cette connexion. Veuillez noter que :

- En général, une station est ainsi reliée à deux autres stations sur une même ligne : la station "avant" et "après" sur la ligne.
- En général, si une arête (a;b) existe (reliant la station a vers la station b), l'arête (b;a) existe également, mais ce n'est pas toujours le cas. Certaines connexions ne se font en effet que dans un seul sens (la boucle de la ligne 10 ou de la ligne 7bis). Les arêtes sont donc orientées (ce sont donc des arcs), et votre structure de données devra respecter les orientations du fichier de données.
- Certaines lignes de métro se dédoublent, comme la fourche de la ligne 13. Le fichier de données en tient compte.
- Une connexion n'est pas seulement caractérisée par une station d'origine et une station de destination, mais également par un numéro de ligne. En effet, deux stations peuvent être liées par deux lignes différentes (par exemple, Grands Boulevards et Bonne Nouvelle sont toutes deux reliées par la ligne 9 et la ligne 8). Voilà pourquoi, dans le fichier de données, chaque arête est caractérisée par deux stations et un numéro de ligne.
- Un "numéro" de ligne n'est pas vraiment un numéro, car certaines lignes peuvent avoir des lettres dans leur numéro.

### 3 L'objectif de votre programme

Votre programme devra proposer tout d'abord un menu permettant à l'utilisateur d'entrer le nom d'une station de départ et d'une station de destination. Vous pouvez, par exemple, proposer une liste de toutes les stations (triées par ordre alphabétique, par ligne, ...) avec des numéros à côté, et l'utilisateur entrera le numéro de la station de son choix. C'est une proposition mais vous êtes libre de choisir la solution qui vous semble la meilleure pour permettre à l'utilisateur d'entrer une station de départ et une station d'arrivée.

Votre programme devra ensuite calculer le trajet le plus court entre les deux stations, en tenant compte qu'il faut 1 minute pour parcourir la connexion directe entre deux stations, et 5 minutes pour un changement de ligne. Ces valeurs devront être représentées par des constantes aisément modifiables dans votre programme.

Enfin, le temps de trajet et le trajet devront être affichés à l'utilisateur. A vous de décider la meilleure façon d'afficher le trajet (sous la forme d'une simple liste des stations à parcourir, sous la forme "Ligne X, directionX" et "Changer à X, ligne X, direction X", ...). La façon dont votre résultat est affiché sera jugée pour établir la note finale.

### 4 Ce qui est à rendre

Vous devrez rendre, dans un fichier ZIP, tout le code source de votre programme et, si nécessaire (et que vous en connaissez le fonctionnement), un Makefile.

Vous devrez rédiger un rapport complet sur votre solution, et le rendre en format PDF : description des structures de données adoptées, choix des structures utilisées à différentes étapes de votre programme, extraits de code qui vous paraissent importants d'expliquer dans le rapport, choix de l'algorithme utilisé, ... Votre rapport doit être un journal de bord permettant d'expliquer chacun de vos choix et chacune des difficultés rencontrées. N'hésitez pas à bien regarder sur Moodle le guide de rédaction d'un rapport, afin de comprendre ce qui est attendu de vous. Votre rapport devra également expliquer comment compiler et utiliser votre programme.

Vous devrez donc rendre un fichier ZIP et un fichier PDF (qui ne devra pas être dans le zip). Tous vos diagramme devront être intégrés dans le PDF (pas de fichiers à part).

**La note de votre projet sera le minimum entre votre note de rapport et votre note de code.** Ne sous estimez donc pas le rapport au profit du code.

Le projet peut être réalisé par groupe de deux au maximum.

Pour information, tous les codes seront comparés pour détecter les plagats. Vous avez le droit de vous aider entre vous, mais pas de recopier le code des autres... Il est fortement déconseillé de partager son code source avec d'autres.