



Devoir 4 (7.5%)

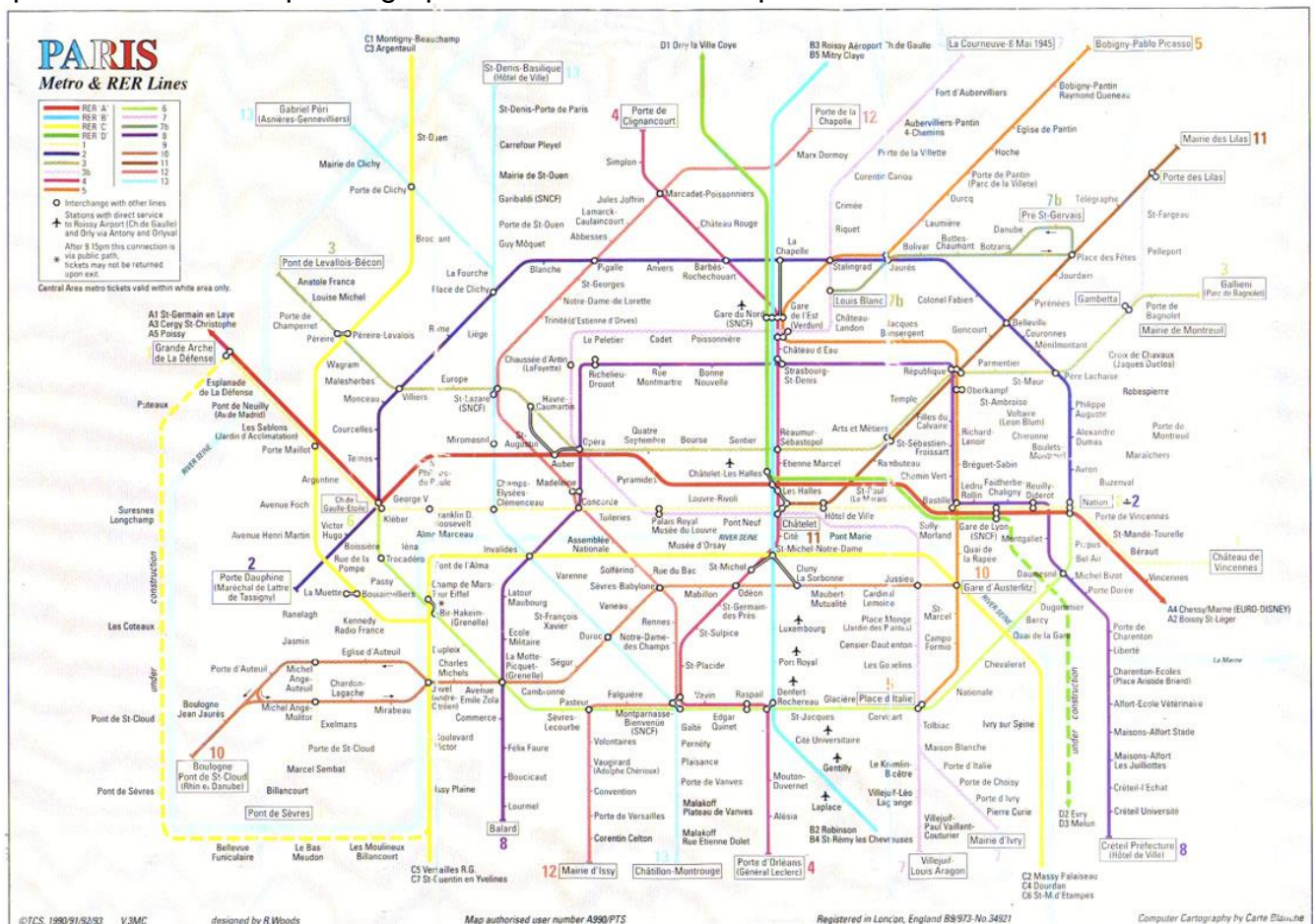
CSI2510

A remettre sur le Campus Virtuel avant 23h55 le 5 décembre 2017

(Pénalité de 30% pour remise tardive de 1 min à 24 heures)

Balade dans le métro Parisien

L'objectif de ce devoir est simplement de trouver le plus court chemin dans un graphe. En attendant la venue du train léger à Ottawa, nous allons considérer le métro de la Ville de Paris qui est un bon exemple de graphe avec une belle complexité.



Le Graphe :

L'information requise pour bâtir le graphe correspondant à ce réseau de métro se trouve dans le fichier texte ci-joint. Ce fichier est organisé comme suit :

- Le fichier débute par deux nombre entiers spécifiant le nombre de sommets (stations de métro) et le nombre d'arêtes.
- Puis chacune des stations sont listées et associées à un numéro. Il faut noter que la même station peut apparaître plus d'une fois lorsqu'elle se trouve sur plus d'une ligne de métro.
- Le symbole \$ signale la fin de la liste des stations.
- Le reste du fichier fournit la liste des arêtes, c'est-à-dire la connexion entre deux stations et le temps qu'il faut (en secondes) au métro pour passer d'une station à l'autre.
- Ce graphe est orienté afin de tenir compte de quelques cas de quelques cas de sens unique.
- Dans certains cas, il est aussi possible de transférer d'une ligne à l'autre en marchant. Ces cas se trouvent à la fin du fichier et sont identifiés avec une pondération de -1.

Questions :

1. Vous devez lire ce fichier et produire un graphe en utilisant la représentation de votre choix.
 - i) Vous devrez donc créer une méthode statique `readMetro(filename)` de lecture
 - ii) Créer une classe `Metro` contenant le graphe du métro
2. A partir de ce graphe, vous devez construire un logiciel permettant de automatiquement:
 - i) Identifier toutes les stations appartenant à la même ligne qu'une station spécifiée. Ces stations doivent être listées dans l'ordre de parcours (incluant la station spécifiée) avec les deux stations aux extrémités de la ligne se trouvant au début et à la fin de cette liste.
 - ii) Trouver le trajet le plus rapide entre deux stations.
 - iii) Trouver le trajet le plus rapide entre deux stations lorsque l'une des lignes est hors-fonction (cette ligne étant identifiée par l'une de ses extrémités).

Afin de spécifier une station, vous devez utiliser son numéro associé. Dans le calcul du temps de parcours, utiliser 90 secondes comme temps de transfert lorsqu'on passe d'une ligne à une autre.

Donner le trajet en listant toutes les stations dans l'ordre et donner la durée totale du trajet.

Votre programme :

Afin d'exécuter votre programme, l'appel doit être comme suit :

```
Java ParisMetro N1 N2 N3
```

Avec N1, N2 et N3 des nombres identifiant des stations de métro.

- Si un seul nombre est spécifié alors le programme fournit la réponse à la question 2i) avec N1 comme station identifiant une ligne à décrire.
- Si deux nombres sont spécifiés alors le programme fournit la réponse à la question 2ii) avec N1 et N2 étant les stations de départ et d'arrivée.
- Si trois nombres sont spécifiés alors le programme fournit la réponse à la question 2iii) avec N1 et N2 étant les stations de départ et d'arrivée et N3 une station à l'extrémité d'une ligne en panne.

Vous devez soumettre votre code en Java ainsi qu'un document décrivant:

- Votre structure de données (classes, attributs and méthodes).
- Description générale des algorithmes utilisés afin de répondre aux questions de la partie 2.
- Exemples de résultats obtenus pour les questions de la partie 2.

Barème de correction:

30% Description des Algorithmes

(clarté des explications, des schémas, bon niveau d'abstraction)

30% Exactitude des solutions

40% Programmation et design

(commentaires, entêtes, clarté et lisibilité du code, longueur des méthodes,

avez-vous utilisé le bon algorithme?, avez-vous défini les bonnes méthodes?, est-ce que vos algorithmes résolvent le bon problème? Votre solution est-elle inutilement trop complexe?, etc)

Remarques importantes:

Pour ce devoir, vous pouvez utiliser n'importe quel code trouvé en ligne ou provenant de nos laboratoires. Vous devez toutefois donner la référence à toute source externe que vous avez utilisé (e.g. le lien menant au code que vous avez téléchargé). Même si vous avez par la suite modifié ce code, ou que vous avez simplement consulté le code pour vous en inspirer, vous devez en donner la référence.

Ne pas fournir les références aux documents et code source que vous avez consulté pourrait être considéré comme de la **fraude scolaire** et donc avoir des conséquences sérieuses.

De même, ce devoir est individuel, à vous de vous assurer que personne n'utilise votre code pour concevoir sa solution. Vous pouvez en discuter en amis, mais s'échanger du code source peut aussi mener à une accusation de **fraude**. Notez que toutes les solutions rendues seront soumis à une application permettant de mesurer la similarité dans la structure de programmes logiciels.

Référence

Devoir inspiré de :

Michel Couprie, Gilles Bertrand, 'Graphes et algorithmes - TP 3',

<https://perso.esiee.fr/~coupriem/Graphestp3/graphestp3.html>, accédé le 14 nov. 2017.