

Programmation Web

Devoir Maison : Node.js : Calcul de trajets en métro

Vous avez tou.te.s sûrement déjà pris le métro dans Paris, probablement même vous le prenez tous les jours pour vous rendre à l'université !

L'objectif de ce devoir est de réaliser une application utile de calcul de trajet sur le réseau de métro de la RATP dans l'agglomération parisienne. Votre application doit permettre de proposer un parcours optimal entre deux stations du métro de Paris.

I) Instructions

1. Ce projet est à effectuer seule.e, en autonomie. **Aucun partage de code n'est toléré.**
2. Ce projet est à rendre pour le 17 avril 2017 à 23h55. Le rendu se fera sur la page Moodle du cours ¹.
3. Le projet devra utiliser le framework Javascript **Node.js**, et pourra inclure des extensions de base comme *express*, *body-parser* etc. mais pas d'extension qui ferait une partie du travail à votre place !
4. Nous n'allons pas juger des qualités artistiques de votre application, mais utiliser un minimum de CSS afin de lui donner un thème visuel pour rendre le tout ergonomique et fonctionnel est une bonne idée. L'utilisation de Bootstrap est autorisée.
5. Nous décrivons chaque fonctionnalité a minima pour vous laisser une certaine liberté ; le fait d'améliorer ce qui est proposé ou de rajouter d'autres fonctionnalités sera pris en compte dans la note.

II) Graphe

Le réseau du métro de la RATP vous est donné sous la forme d'un graphe entre stations. La première étape consiste à importer le graphe : lire les fichiers fournis et encoder le graphe dans des structures Javascript, afin de pouvoir appliquer un calcul de plus court chemin.

Ce graphe est orienté et pondéré : il contient des sommets (les stations), des arcs (ces deux stations sont voisines sur une même ligne de métro, ou en correspondance), et des poids pour les arcs. Seules les stations de métro sont incluses, les stations de RER et tram ne sont donc pas présentes. Une visualisation de ce graphe, incarnée par le plan du métro parisien, est très connue et disponible dans toutes les stations de métro de Paris. Une version libre de Wikipédia est reproduite page suivante.

Format du graphe Un point essentiel à connaître en premier : les stations sont représentées par des numéros, et les stations permettant des correspondances sont représentées par plusieurs numéros. Cela veut dire que la station Bastille par exemple, qui est desservie par 3 lignes, porte 3 numéros différents : un pour chaque ligne desservie.

Le graphe fourni comprend 2 fichiers principaux. Dans chaque fichier, les données sont en colonnes, séparées par des espaces. Ces fichiers sont au format UNIX : les lignes sont séparées simplement par des `\n` (attention, cela peut causer des surprises si vous les ouvrez sous Windows !)

1. <http://moodlesupd.script.univ-paris-diderot.fr/course/view.php?id=6737>

- `metro_graphe.labels` : les numéros affectés à chaque station pour chaque ligne, entre 0 et 382, ainsi que le nom correspondant.

Exemple : *42 Bréguet-Sabin* signifie que la station 42 dans le fichier `.edges` est Bréguet-Sabin. Le numéro 18 correspond à Bastille-ligne 1, le numéro 19 à Bastille-ligne 5 et le 20 à Bastille-ligne 8.

- `metro_graphe.edges` : la liste des tronçons entre stations, ainsi que des correspondances, permettant de déduire les arêtes du graphe, au format `[station depart] [station arrivee] [poids]`.

Il existe un arc par tronçon par sens, avec un poids de 1. La plupart des arcs sont donc dédoublés car les métros parcourent les stations dans les deux sens, mais il existe des exceptions, comme les boucles à sens unique de la ligne 7bis ou de la ligne 10!

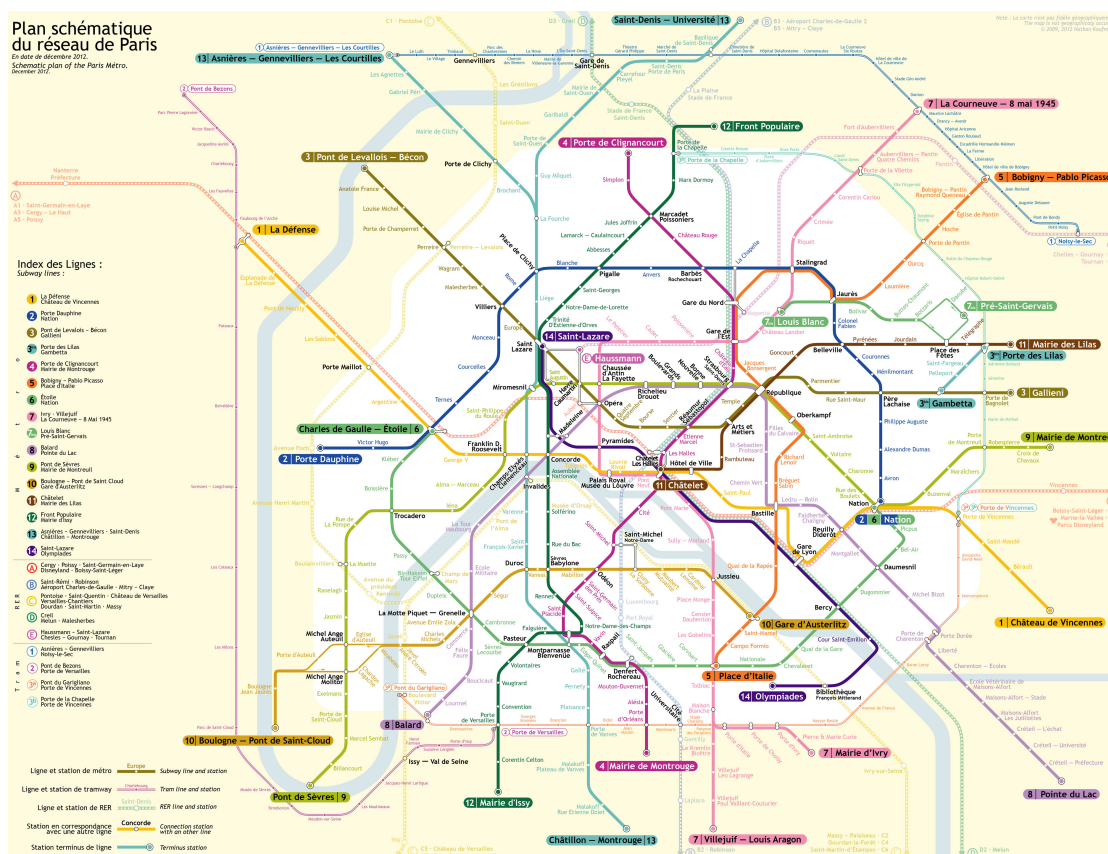
Un arc entre deux stations portant le même nom signifie une correspondance, et a un poids de 2. Ces arcs sont présents en fin de fichier, à partir de la ligne 728.

Par exemple, *21 96 1* signifie que les stations *Bel-Air* et *Daumesnil-ligne 6* sont reliées par le même métro (la ligne 6 en l'occurrence). Il existe un arc réciproque *96 21 1*.

Il existe un arc *95 36 1* (Danube → Botzaris) pour la ligne 7bis, mais pas de *36 95 1* car Danube est dans une boucle à sens unique! (regarder le plan)

L'arc *118 119 2* signifie qu'il existe une correspondance entre les deux lignes desservant la station *Gambetta*.

Pour vous aider, vous disposez également d'un fichier par ligne de métro (`metro_ligneX.stations`), contenant les numéros de chaque station de la ligne, au format un numéro par ligne. Attention, ne comptez pas seulement sur ces fichiers, en effet, toutes les stations, bien qu'indiquées consécutivement, ne sont pas nécessairement reliées (cas des boucles, des fourches). Le vrai travail est à effectuer avec le fichier `metro_graphe.edges`! Réfléchissez à comment détecter une correspondance dans le chemin renvoyé par l'algorithme de Dijkstra, par exemple.



III) Algorithme de Dijkstra

Afin de proposer un parcours dans le réseau ferré RATP, vous aurez besoin d'un algorithme de calcul de plus court chemin. Nous vous conseillons pour cela un des algorithmes les plus connus et les plus utilisés : l'algorithme de *Dijkstra*². De nombreuses descriptions et pseudo-codes de cet algorithme sont disponibles sur Internet, la page Wikipédia est, par exemple, un bon début.

Le graphe étant orienté, chaque arc ne peut être parcouru que dans un seul sens, les tronçons disponibles dans les deux sens étant dédoublés. Par exemple un arc nommé $a\ b$ dans le fichier vaut pour $(a \rightarrow b)$. S'il n'est pas dans une boucle, ou que ni a ou b est un terminus, alors il y aura normalement un arc $b\ a$ pour $(b \rightarrow a)$ également présent dans le fichier.

Le graphe étant pondéré en fonction des correspondances, votre algorithme devrait naturellement pénaliser un changement de ligne de métro.

IV) Fonctionnalités minimales attendues

Les fonctionnalités ci-dessous garantissent la moyenne si elles sont implémentées correctement. Mieux l'implémentation sera faite, plus la note sera élevée. Les meilleures notes ne seront atteintes qu'avec la réalisation de fonctionnalités optionnelles.

Saisie d'une station de départ, d'une station d'arrivée Votre interface doit permettre de saisir une station de départ ainsi qu'une station d'arrivée, au format texte.

Calcul d'un parcours Vous devez pouvoir proposer un parcours entre n'importe quels points de départ et d'arrivée, même non-optimal³, avec moins de 4 changements. Le graphe étant fortement connexe, aucun parcours n'est impossible.

Affichage du parcours (correspondances incluses) Vous devez afficher votre parcours segment par segment, en indiquant à l'utilisateur quand il doit changer de ligne.

Gérer les erreurs Votre programme doit afficher une erreur et rester fonctionnel même si l'utilisateur effectue des saisies incorrectes.

V) Fonctionnalités optionnelles

Implémenter une ou plusieurs des fonctionnalités supplémentaires suivantes vous permettra de maximiser votre note.

Elles sont classées par ordre de difficulté (de ce qui nous semble le plus facile, au plus difficile).

Proposer une estimation du temps de parcours Demandez une heure de départ à l'utilisateur, estimez son heure d'arrivée ainsi que son temps de trajet. Dans un premier temps, vous pouvez prendre les temps moyens suivants : 1.2 minutes entre chaque station, 20 secondes à quai, 1.5 minute pour une correspondance. Vous pouvez affiner ces temps en cherchant des temps de référence pour les correspondances sur Internet, si vous le souhaitez.

Suggestion des noms de station Utilisez de l'auto-complétion pour les champs de texte afin de suggérer les noms des stations de départ et d'arrivée.

Proposer des corrections de nom après soumission Petit corollaire du point précédent, si le nom d'une station soumis par l'utilisateur est introuvable en tant que tel, essayez de proposer des corrections du nom de station.

2. Nom néerlandais, prononcer Déijkstra

3. Si vous avez implémenté l'algorithme de Dijkstra correctement, le parcours sera optimal.

Impact de la pénalisation des changements Proposez à l'utilisateur de choisir la pénalisation à appliquer aux changements, et répercutez-la sur les poids des arcs du graphe.

Direction de la ligne de métro à prendre En ajoutant des informations additionnelles au graphe, indiquez lors d'un changement la direction de la ligne à prendre (exemple : *Prendre la ligne 14 direction Saint-Lazare*).

VI) Références de comparaison

Vous pouvez bien sûr vous comparer avec des sites existants, en premier lieu :

- <http://www.ratp.fr/itineraires/>
- <http://www.vianavigo.com/>

N'oubliez pas de sélectionner dans les critères d'utiliser uniquement le métro !

Ces sites ont des informations plus fines sur les temps de correspondance et les horaires des passages qui peuvent influencer sur les temps de trajet, mais globalement vous devriez retrouver à peu près les mêmes résultats !