Rapport projet INE 403 algorithmique 1
"METRO project"

M.ELHAMIM Charafedine - 21807418

M.BELHANNACHI moundji - 21921430

1 Exécution

-Pour exécuter le projet il suffit de taper la commande "make", ceci est valable pour les deux versions.

2 Contenu

Nous avons rajouté quelques données dans le fichier 'metro.txt'. En effet, il était plus intéressant de connaître les lignes de chaque station. Les terminus pour chaque arc étaient également une donnée intéressante. Le projet a deux versions :
-Version 1 console qui ne fais qu'afficher les instructions et la durée du trajet.
-Version 2 interface qui en plus d'afficher les instructions et la durée du trajet elle affiche aussi la carte du réseau de métro parisien avec possibilité de zoomer.

2.1 Version 1 console

Fichiers code source et données :
- 6 fichiers '.java' dans le répertoire 'src'
- un fichier 'metro.txt' dans le répertoire 'ressources'
pour connaître les directions.

2.2 Version 2 interface

Fichiers code source et données :
- 7 fichiers '.java' dans le répertoire 'src'
- un fichier 'metro.txt' dans le répertoire 'ressources'
- deux répertoires 'css' et 'view' qui concerne l'interface.

3 Objectifs

-afficher le chemin qui prend le moins de temps d'une station de métro A à une station de métro B dans le réseau des lignes de métro parisien.
-afficher les instructions à suivre afin d'arriver à la station B,il sera indiqué dans les instructions : les stations de métros à emprendre et la direction à suivre.
-afficher la durée du trajet.

4 Représentations et implémentions

Le projet est entièrement programmer en 'JAVA' sauf la partie interface qui utilise 'JAVAFX', 'CSS' et 'xml'.

La class 'Node' représente une station de métro et elle contient :
- l'indice dans le fichier,le numéro de ligne et le nom d'une station et tous ces attributs sont récupérer du fichier 'metro.txt'.
- les getters et les setters des attributs qui nous permettent de récupérer la valeur d'un attribut ou de le modifier.
- la méthode 'add edge' qui permet d'ajouter une connexion entre deux stations voisines.
- la méthode 'reset' qui permet de réinitialiser les structures de données nécessaire au calcul du chemin qui prend le moins de temps.

La class 'Edge' représente une connexion entre deux stations et contient :
-une station de départ de type 'Node',une station d'arriver de type 'Node' et la distance du trajet entre les deux stations.
-les getters et les setters des attributs qui nous permettent de récupérer la valeur d'un attribut ou de le modifier.

La class 'graph' représente le réseau des lignes de métro parisien et elle contient :
- une liste chainée 'ArrayList' de 'Node' au sont stockées les stations du réseau.
- la méthode 'addNode' qui ajoute une station au réseau. - les deux occurences de la méthode 'getNodes' sert à retourner une station du réseau selon l'indice ou le nom. - la méthode 'getList' qui va retourner la liste des stations du réseau.
- la méthode 'reset' qui sert à réinitialiser des attributs dans la classe 'Node'
- la méthode 'ligneB' qui sert à passer d'un numéro de ligne dans le fichier 'metro.txt' à un numéro de ligne réel. - la méthode 'getPath' qui sert à afficher les instructions à suivre et la durée du trajet.

La class 'ParseMetro' contient :
- la méthode 'initGraphe' qui va initialiser les attributs de la classe 'Graph'

5 Djikstra

A première vue, l'algorithme qui était le plus adapté pour calculer le plus court chemin entre des stations de metro était celui de Floyd-Warshall car il calcule entre toutes les paires de sommets. Pourtant, sa complexité est de o(n^3) avec n le nombre de sommets. Mais, l'algorithme de Dijkstra a une complexité de o(n^2) : à chaque itération on traite un sommet o(n) et à chaque traitement de sommet, on parcourt le tableau des plus courtes distances o(n). Et s'il fallait atteindre la même que Floyd-Warshall, il aurait fallu n fois chemins différents o($n^2 * n$) : ce qui n'est pas réaliste pour 375 stations. C'est pour cette raison que nous avons choisi Dijkstra pour calculer le plus court chemin entre stations.

6 Remarques

-Pour exécuter la version 2 interface il essentiel d'avoir la version JAVA 11 ou une version plus récente installé sur la machine.
-Lors de l'exécution de la version 2 interface l'interface peut prendre quelques secondes avant de s'afficher sur l'écran.

7 détails interface inachevé

Afficher sur la carte le chemin à suivre.