Polytechnique Montréal

Département de génie informatique et génie logiciel

Cours LOG2810:

Stuctures Discrètes

Travail pratique 1

**Graphes**

Par l'équipe

No 2

Noms:

Alexander Ciaciek ???????

Nour Abiad ??????

Dana Louka 1943144

Date:

22 Mars 2020

Introduction:

Ce laboratoire demande la création d’un programme fonctionnelle qui ressemble à l’application Uber. Les clients doivent pouvoir utiliser ce service en ayant des contraintes temporelles. Le conducteur doit utiliser le chemin le plus rapide sans que la batterie de son voiture électrique baisse en-dessous de 15%. Afin d’arriver à ce but, les notions de graphes (chemins, sommets, arrêtes), les structures de données (les listes chainées en particuliers) et l’algorithme Dijkstra ont été utilisés. Premièrement, les demandes du laboratoire ont été analysés et ont permis la conception des différentes classes. Deuxièmement, les méthodes ont été codés afin de pouvoir lire un document texte passé en argument récursivement, de trouver le chemin le plus court dans un graphe et de profiter et parcourir les sommets/ les arrêts d’un graphe. Enfin, une interface qui affiche un menu a été mise en place.

Solution :

Figure 1 : Diagramme de classes.

C’est possible de remarquer que dans la classe « Graphe » est composée de sommets. Chaque sommet contient les informations d’un arrondissement, puisque c’est ce qu’un sommet représente : un ID, l’existence d’une station de recharge, une liste d’arcs adjacents. Un arc est composé d’un Sommet et un poids (ou un temps), ce qui fait que le graphe soit pondéré.

La classe Main, contient une méthode afficherOptions(). Celle-ci se charge d’afficher une interface simple. La première option permet d’afficher le graphe lu. La deuxième se charge de trouver un le plus court chemin entre sommet Départ et sommet Destination. La troisième analyse les requêtes des clients. Une boucle while et un switch permettent de choisir entre ces options.

blabla

Difficultés :

La chose la plus difficile à faire dans ce laboratoire est de faire le lien entre les structures discrètes, comme le graphe, et l’implémentation en code avec des structures de données. En effet, le choix des listes comme structure de données au lieu d’un hashMap, ou autre structure a été difficile à prendre. Cependant, les listes sont faciles à parcourir et peuvent bien sauvegarder toutes les informations d’un sommet. Plusieurs de leurs méthodes sont de complexité O(1) et, en plus, un algorithme de Dijkstra ou d’affichage oblige de parcourir tout le graphe, donc c’est impossible de faire mieux qu’une complexité O(n).

Une deuxième difficulté a été de bien utiliser les librairies du langage Java, la langue de programmation utilisé dans ce laboratoire. Il y avait plusieurs classes et méthodes qui ont facilité le travail. Il fallait découvrir en soi-même ces méthodes pertinentes. Par exemple, la classe Comparator a simplifié énormément la méthode trouverSommetAvecPoidsMinimal() afin de trouver le poids minimal.

Conclusion :

Ce laboratoire était utile car ils permettent d’approfondir les connaissances sur les fonctions récursives, la lecture des fichiers et l’écriture d’un programme complet à partir de rien. Ce laboratoire a pris beaucoup de temps, car il fallait se familiariser avec le langage de programmation, l’algorithme de Dijkstra et la conception des classes. Il fallait refaire plusieurs fois les diagrammes de classe et écrire beaucoup de pseudocode. En plus, il fallait fouiller l’internet pour connaitre la syntaxe et les classes intéressantes en Java. Le prochain laboratoire serait aussi demandant et intéressant. Il faut mettre beaucoup de temps afin d’arriver.