

# Projet Structures de données : avancé

## 1 Objectif

Les fichiers `aeroports.txt` et `vols.txt` contiennent des données à propos du trafic aérien. `aeroports.txt` contient des informations sur des aéroports et `vols.txt` sur des vols.

Pour ce projet, nous vous demandons (d'implémenter un programme java qui permettra de calculer un itinéraire pour aller d'un aéroport à un autre. Vous devrez implémenter deux façons de calculer un itinéraire :

1. qui minimise le nombre de vols,
2. qui minimise le nombre de kilomètres parcourus.

Votre programme affichera les itinéraires calculés à la console. Ci-dessous, voici un exemple d'itinéraire minimisant le nombre de vol entre l'aéroport de Bruxelles et l'aéroport de Christchurch International Airport. Cet itinéraire est composé de 3 vols et passe par Helsinki et Singapore.

```
distance : 19324.185556164022
Vol [source=Brussels Airport, destination=Helsinki Vantaa Airport, airline=Finnair,
distance=1646.4344602223553]
Vol [source=Helsinki Vantaa Airport, destination=Singapore Changi Airport, airline=Finnair,
distance=9274.90670132227]
Vol [source=Singapore Changi Airport, destination=Christchurch International Airport,
airline=Singapore Airlines, distance=8402.844394619397]
```

Pour un itinéraire, on affiche simplement la distance en km et ensuite les vols contenus dans l'itinéraire.

Il est peut-être possible de trouver d'autres itinéraires composés de 3 vols qui relie Bruxelles à Christchurch.

Voici un exemple d'itinéraire minimisant le nombre de km parcouru en avion entre Bruxelles et Christchurch. Cet itinéraire passe par Sheremetyevo, Guangzhou et Brisbane.

```
distance : 18810.832866785186
Vol [source=Brussels Airport, destination=Sheremetyevo International Airport,
airline=Aeroflot Russian Airlines, distance=2229.5122705405697]
Vol [source=Sheremetyevo International Airport, destination=Guangzhou Baiyun International
Airport, airline=China Southern Airlines, distance=7002.741156164331]
Vol [source=Guangzhou Baiyun International Airport, destination=Brisbane International
Airport, airline=China Southern Airlines, distance=7080.850125307775]
Vol [source=Brisbane International Airport, destination=Christchurch International Airport,
airline=Singapore Airlines, distance=2497.7293147725118]
```

On remarque bien que ce deuxième itinéraire contient plus de vols (4 au lieu de 3) mais parcourt une distance moindre (18810 km au lieu de 19324 km).

## 2 Sur Moodle

Pour mener à bien votre projet, nous vous fournissons plusieurs fichiers :

- `aeroports.txt` contient des informations à propos de 5636 aéroports. Il contient une ligne par aéroport. Pour chaque aéroport, les informations suivantes sont séparées par des virgules dans cet ordre : le code IATA qui identifie l'aéroport, le nom de l'aéroport, la ville, le pays, la longitude et la latitude.
- `vols.txt` contient des informations à propos de 63570 vols. Il contient une ligne par vol. Pour chaque vol, les informations suivantes sont séparées par des virgules dans cet ordre : le nom de la compagnie aérienne, le code IATA de l'aéroport source et le code IATA de l'aéroport de destination.
- la classe `Util.java` possédant une méthode statique permettant de calculer la distance entre deux coordonnées géographiques. Une coordonnée géographique est composée d'une longitude et d'une latitude.
- une classe `Main.java` que vous ne pouvez pas modifier. Le code de cette classe est présenté ci-dessous. La classe à construire `Graph` devra avoir un constructeur qui prendra les deux fichiers contenant les informations des aéroports et des vols en paramètre. Elle devra également contenir deux méthodes pour calculer un itinéraire. Ces deux méthodes prennent deux paramètres qui sont les codes IATA des aéroports source et destination.

```
import java.io.File;

public class Main {
    public static void main(String[] args) {
        try {
            File airports = new File("airports.txt");
            File vols = new File("vols.txt");
            Graph g = new Graph(airports, vols);
            g.calculerItineraireMinimisantNombreVol("BRU", "CHC");
            System.out.println("-----");
            g.calculerItineraireMinimisantDistance("BRU", "CHC");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

### 3 Tâches à effectuer

Nous vous demandons de rendre la classe `Main` fonctionnelle en réalisant les différentes tâches suivantes :

- Implémenter le constructeur de la classe `Graph` qui lira les deux fichiers contenant les aéroports et les vols.
- Implémenter les deux algorithmes pour calculer les itinéraires en utilisant la théorie sur les graphes. S'il est impossible d'aller de l'aéroport source à l'aéroport destination, votre programme lancera une exception.

### 4 Tâches bonus

Si vous avez fini trop tôt et que vous vous embêtez durant les séances, vous pouvez faire la tâche bonus. L'objectif est de faire le projet durant les séances, ne prenez pas du temps à la maison pour faire cette tâche bonus ; privilégiez les autres cours comme le projet PAE. On ne s'attend pas à que vous fassiez cette tâche bonus.

Voici la tâche bonus : Ajoutez une troisième façon de calculer un itinéraire en minimisant le nombre de compagnies aériennes différentes utilisées dans un itinéraire. Si deux itinéraires utilisent le même nombre de compagnies, privilégiez l'itinéraire parcourant le moins de km.

## **5 Organisation et livrables**

Ce projet se déroule du 9 mars 2022 au 27 mars 2022 par groupe de deux étudiants. Les étudiants d'un même groupe doivent être dans la même série. Durant tout le projet, la présence aux cours est obligatoire.

Si une série comporte un nombre impair d'étudiant, nous accepterons un seul groupe de 3 étudiants.

Le projet est à remettre via « Moodle » pour le dimanche 27 mars 2022 à 12h00 (midi). Nous ne demandons pas de rapport. Si certaines choses méritent des explications, vous pouvez les fournir en commentaire dans les différents fichiers.

**Le plagiat sera lourdement sanctionné (au minimum 0 à l'UE). Tous les projets seront automatiquement testés par un logiciel de détection de plagiat.**