IUT Lyon 1
l'excellence technologique

SAE S2 2023-2024

PLANIFICATION DU TRANSPORT AÉRIEN

3 SAE en une

- **SAE 201 – Développement d’une application**

Partir d'un besoin exprimé par un client nécessitant une interface graphique, formaliser les besoins, proposer une conception, implémenter et tester son développement.

- **SAE 202 – Exploration algorithmique d'un problème**

Approfondir la réflexion sur l'approche algorithmique des problèmes rencontrés pendant les phases de développement. Algorithmes en lien avec les graphes.

- **SAE 205 – Gestion de projet**

Analyser les besoins de l'entreprise et rédiger un cahier des charges. Première familiarisation avec la conduite de projet à travers un sujet simple.

Ressources concernées

Java : 4h

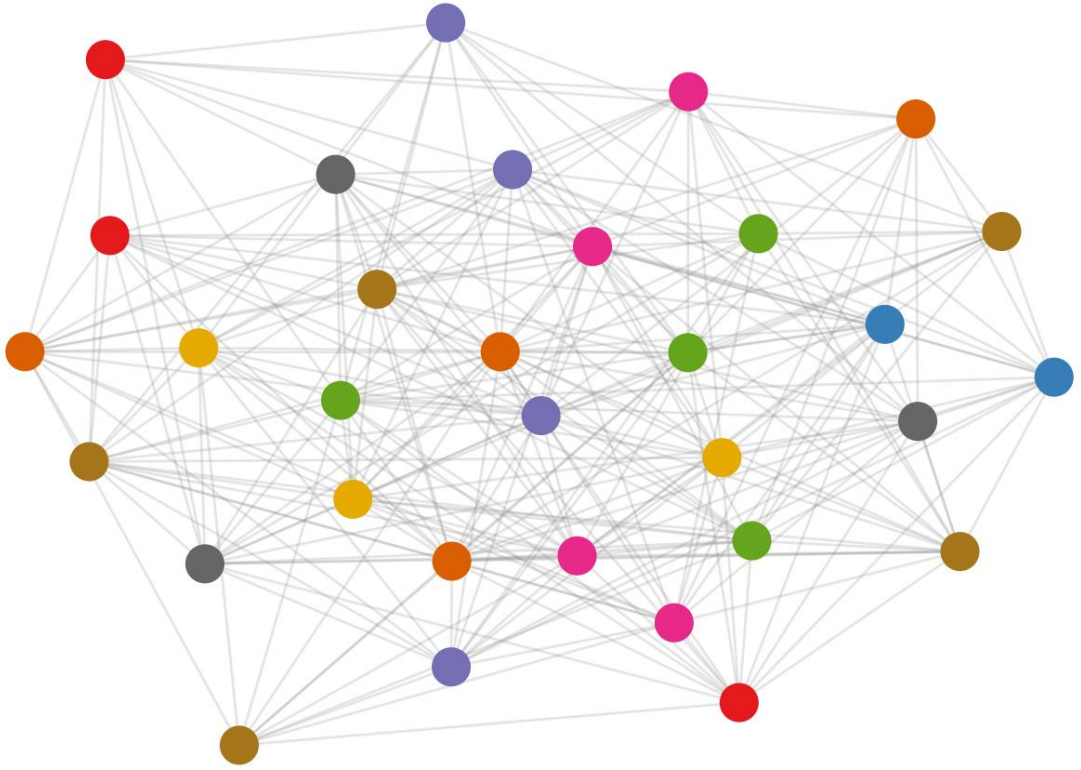
IHM : 10h

Graphes : heures de cours

Qualité de développement : 6h

Gestion de projet : heures de cours

SDD : 4h



Description

Vous devez gérer l'espace aérien français pour un ensemble de vols entre plusieurs aéroports.

Un vol est caractérisé par son aéroport de départ, son aéroport d'arrivée, son heure de départ et sa durée.

Deux vols sont susceptibles de rentrer en collision si:

- leurs trajectoires dans l'espace s'intersectent

- l'écart entre les horaires de passage théoriques de chacun des deux avions au point d'intersection n'excède pas 15 minutes (<15mn).

On négligera la forme sphérique de la Terre (on considèrera les trajectoires dans le plan)

Description

Pour éviter les collisions, deux vols susceptibles de rentrer en collision vont voler à des niveaux différents. On numérotera par 1,2, … *kmax* les différents niveaux de vols possibles.

kmax est une constante du problème qui correspond au nombre maximum de niveaux autorisés.

On souhaite trouver une affectation avec au plus *kmax* niveaux sur l'ensemble des vols d'une journée.

N.B: Dans certains cas, il ne sera pas possible de trouver de solution si *kmax* est trop petit.

Modélisation

Le problème peut se modéliser comme un problème de coloration de graphe.

A partir de la liste des vols et des aéroports, on peut construire le **graphe d'intersection des vols**:

- un sommet est un vol

- deux sommets sont adjacents si les deux vols correspondant sont susceptibles de rentrer en collision.

Trouver une affectation des niveaux de vols revient à **colorier le graphe avec au + *kmax* couleurs.**

Objectif

Dans certains cas, il ne sera pas possible de proposer une coloration légale avec *kmax* couleurs.

Ex: si *kmax*=2 et le graphe n'est pas 2-coloriable.

Votre objectif sera donc de **colorier le graphe en minimisant le nombre de conflits**.

Un conflit est une arête dont les deux extrémités sont coloriées avec la même couleur. Dans le cas où vous réussissez à colorier le graphe sans conflit, cela signifie que vous avez trouvé une coloration avec *kmax* couleurs.

Vous pourrez utiliser différents algorithmes de coloration.

Travail demandé

1. **Construire le graphe d'intersection des vols**

- à partir du fichier .csv avec la liste des aéroports (cf. Moodle) formaté de la façon suivante: sur chaque ligne on a les informations d'un aéroport

```
AGF Agen 44 10 29 N 0 35 26 E
```

où AGF est le code de l'aéroport, Agen est le lieu, et le reste correspond à latitude et longitude (degré/minute/seconde/orientation)

- à partir d'un fichier .csv de vols

Il y aura plusieurs fichiers de vols intitulés vol-testX.csv, formatés ainsi : sur chaque ligne on a un vol

```
AF98022 LME MLH 8 29 47
```

où AF98022 est le nom du vol, LME l'aéroport de départ, MLH celui d'arrivée, départ 8H29, durée: 47 minutes

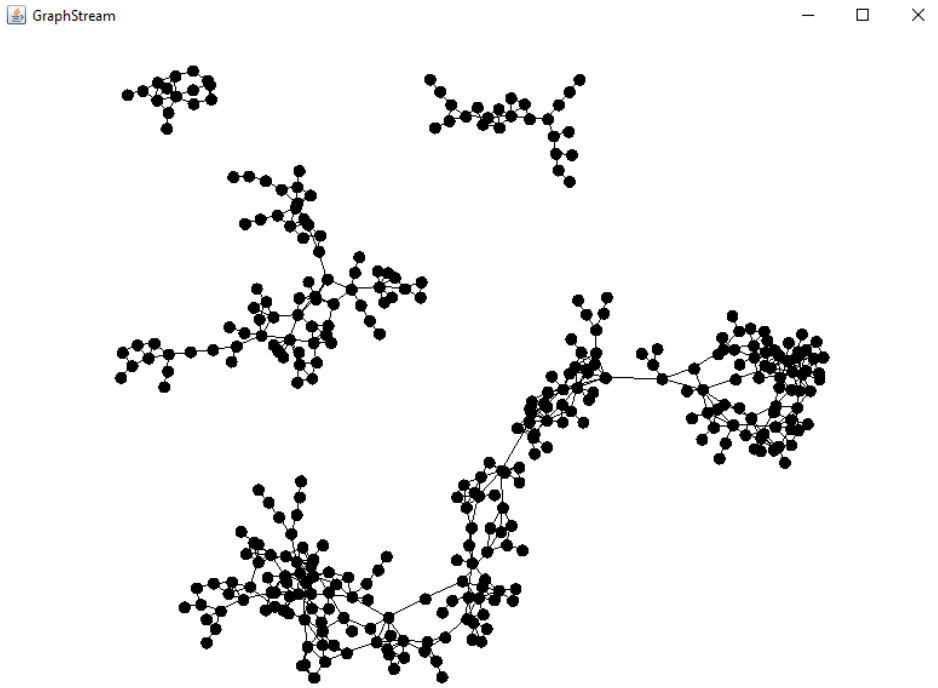
Travail demandé

2. **Visualiser le graphe d'intersection d'un fichier de vols**

Utilisation de la librairie **Graphstream** (sources sous Moodle)

https://graphstream-project.org/

Le jour de l'évaluation, construire les graphes d'intersection issus de 5 fichiers de vols différents (vol-evalX.csv) et donner leur nombre de sommets et arêtes.



Travail demandé

3. **Lire et charger un graphe directement depuis un fichier au format (graph-testX.txt)**

Un fichier contenant un graphe sera un fichier d'extension .txt formaté comme suit:

ligne 1: nombre de couleurs maximum (*kmax*)

ligne 2: nombre de sommets du graphe.

toutes les lignes suivantes décrivent les arêtes (une ligne par arête), avec les identifiants des 2 sommets extrémités séparés par un espace.

Travail demandé

4. **Ecrire un/des algorithmes qui colorient un graphe en minimisant le nombre de conflits**

Rappel: vous ne devez utiliser que des couleurs entre 1 et *kmax*.

Vous pourrez implémenter autant d'algorithmes de coloration de votre choix.

Des pistes d'algorithmes sont données dans la doc sous Moodle.

La coloration doit être visualisée sur le graphe via Graphstream (doc Moodle pour cela)

Travail demandé

5. **Une interface graphique avec plusieurs actions et vues possibles. Voici une liste de fonctionnalités souhaitées:**

- charger un graphe directement (fichiers *graph-test.csv*)
- choisir parmi différents algorithmes de coloration et afficher le nombre de conflits
- afficher des stats sur le graphe en cours (degré moyen, nombre de composantes connexes, nombre de nœuds, nombre d'arêtes, diamètre…)
- charger une liste d'aéroports au format csv (fichier *aeroport.csv*)
- charger une liste de vols au format csv (fichiers *vol-test.csv*)
- construire le graphe d'intersection des vols une fois les deux listes précédentes chargées
- afficher/éditer kmax
- afficher le graphe en cours (qu'il provienne d'un fichier de vols ou pas)
- visualiser une coloration du graphe en cours
- zoomer sur le graphe

Travail demandé

6. **Charger une liste de 20 graphes d'évaluation et les colorier avec le moins de conflits possibles.**

Vous aurez 30 min pour colorier les 20 graphes.

Rappel: vous ne devez utiliser que des couleurs entre 1 et *kmax (avec kmax indiqué dans chaque fichier)*

Chaque graphe d'évaluation aura pour nom: *graph-eval0.txt, graph-eval1.txt, … , graph-eval19.txt*

Vous produirez en sortie plusieurs fichiers (que vous déposerez sous Moodle)

- un fichier csv dans le dossier nommé coloration-groupeX.Y.csv contenant sur chaque ligne les informations suivantes: nomFichier.txt;nbre conflits

- pour chacun des 20 fichiers traités, un fichier contenant toute la coloration de votre graphe (noeud par noeud). Chaque fichier s'appelera *colo-eval0.tx, colo-eval1.txt*… Vous mettrez ces 20 fichiers dans une archive portant le numero de votre groupe.
  Le format sera le suivant : pour chaque ligne indiquez *numero du sommet ; couleur*

NB : Cette partie se fait sans visualisation, seul le nombre de conflits doit être récupéré.

Travail demandé

7. **IHM avancée**

- possibilité de changer la marge de sécurité pour les vols
- visualiser les vols et les aéroports (s'ils sont chargés) sur une carte de France (regarder la librairie JXMapViewer)
- afficher tous les vols en cours à une heure donnée
- étant donné une affectation des niveaux de vols, sélectionner un aéroport et visualiser les niveaux de chaque vol
- étant donné une affectation des niveaux de vols, sélectionner un niveau et afficher tous les vols pour ce niveau

Travail demandé

8. **Qualité du code**

- JavaDoc à rendre sous Moodle (vos méthodes publiques doivent pouvoir être réutilisées par qqn d'extérieur)

- Organisation du code/projet en packages, répertoires pour les data etc…

- Tests unitaires à faire

- Gestion des erreurs s'il y a des fichiers erronés

Travail demandé

9. **Cahier des charges**

- A rendre sous Moodle pour le 13 mai, selon les indications de Mme Dufour.

- Consignes données par Mme Dufour. Le Cahier inclura des maquettes pour la partie IHM notamment.

Organisation

- Travail en trinômes

- Déroulement : travailler sur TOUT le projet durant les séances dédiées des modules concernés. L'encadrant peut vous aider pour des questions relatives à sa matière. Une partie s'effectue en totale autonomie hors des séances PE.

Outils utilisables

- Gestion de projet et versionning de code :
  - Trello, GitLab (forge Lyon1)
- Outils de maquettage
  - Figma, Wireframesketcher, Balsamiq Wireframes, https://cacoo.com
- AGL (Atelier de Génie Logiciel)
  - PowerAMC, Astah UML https://astah.net , Visual Paradigm
- IDE (Environnement de Développement)
  - Netbeans sera utilisé en Java et IHM Java, librairie Graphstream

Evaluation

Vous aurez au total 5 notes

N1: Une note /20 pour le cahier des charges (à rendre pour le 13 mai)

N2: Une note /20 de démo devant votre enseignant d'IHM, sur la qualité de l'interface et les fonctionnalités (parties 2, 5 et 7). Semaine du 24 juin.

N3: Une note /15 sur la partie 6 la semaine du 24 juin. Votre note sera calculée automatiquement selon la performance de vos résultats / autres.

N4: Une note /5 de maths pendant le DS de maths.

N5: Une note /20 de qualité (partie 8). Semaine du 24 juin.

Note par SAE

SAE 201

- 2/3 N2 + 1/3 N5

SAE 202

- N3+N4

SAE 205

- 2/3 N1 + 1/3 N5