Master 1 – Modélisation, Graphes et Algorithmes
Projet : calcul d'un circuit eulérien

1 Description du projet

Un *cycle eulérien* est un cycle qui passe par chaque arête d'un graphe exactement une fois. Un graphe connexe admet un cycle eulérien si et seulement si chaque sommet a un degré pair.

Question 1 *Écrire une fonction* `VerifGrapheEulerien` *qui attend en paramètre un graphe et qui retourne* `True` *si et seulement si le graphe est eulérien.*

Question 2 *Écrire une fonction* `TestCycleEulerien` *qui attend en paramètre un graphe $G = (V, E)$ et une liste $\mathcal{C}$ de sommets (la liste $\mathcal{C}$ peut contenir plusieurs fois des mêmes sommets). Cette fonction retourne* `True` *si et seulement si la liste $\mathcal{C}$ est un cycle eulérien de $G = (V, E)$.*

L'objectif de ce projet est d'implémenter un algorithme pour le calcul d'un cycle eulérien. Cet algorithme, datant de 1883, est connu sous le nom d'ALGORITHME DE FLEURY [1]. Avant d'en donner sa description, il nous faut définir la notion d'*isthme*.

Étant donné un graphe $G = (V, E)$, on dit qu'une arête $e = \{x, y\} \in E$ est un *isthme* si la suppression de cette arête e résulte en la séparation des sommest x et y dans deux composantes connexes distinctes.

Question 3 *Écrire une fonction* `EstIsthme` *qui attend en paramètre un graphe $G = (V, E)$ et une arête e. Cette fonction retourne* `True` *si et seulement si l'arête e est un isthme de G.*

Nous donnons ci-après une description de l'algorithme de Fleury :

Algorithme de Fleury($G = (V, E)$, u)
Entrée : un graphe G représenté par listes d'adjacence et un sommet u.
Sortie : un cycle eulérien $\mathcal{C}$ de G commençant et finissant au sommet u.

$\mathcal{C} \leftarrow [u]$ `/* liste contenant initialement que le sommet` u `*/`
$F \leftarrow G$ `/* copie de` G `dans` F `*/`
$SommetCourant \leftarrow u$
tant que *il existe des arêtes incidentes à* SommetCourant *dans le graphe F* **faire**
- Choisir une arête $e = \{SommetCourant, y\}$ incidente à $SommetCourant$ dans F telle que e ne soit pas un *isthme* dans le graphe F, excepté s'il n'y a pas d'autre choix pour l'arête e
- Ajouter le sommet y à la fin de la liste $\mathcal{C}$
- $SommetCourant \leftarrow y$
- $F \leftarrow F \setminus e$ `/* suppression de l'arête` e `dans le graphe` F `*/`

retourner le cycle $\mathcal{C}$

Question 4 *Implémenter l'algorithme de Fleury.*

2 Contraintes d'implémentation

Pour l'implémentation, vous utiliserez le langage de votre choix parmi Python, Caml, C++, Java. Les graphes utilisés dans votre implémentation devront être représentés par des listes d'adjacences.

En utilisant la fonction `TestCycleEulerien`, vous devez vérifier que le cycle construit par votre implémentation de l'algorithme de Fleury est bien un cycle eulérien.

2.1 Entrée

Afin de tester votre algorithme, votre programme devra lire un graphe stocké dans un fichier. Le graphe est donné sous le format suivant qu'il est **impératif de respecter**.

```
n
x11 x12 x13 ...
x21 x22 x23 ...
...
xn1 xn2 xn3 ...
```

où :
- `n` est le nombre de sommets du graphe et les sommets sont numérotés de 1 à n ;
- la ligne `xi1 xi2 xi3 ...` représente la liste des voisins du i-ième sommet.

2.2 Sortie

Votre programme affichera sur la sortie standard un cycle eulérien du graphe donné en entrée, sous la forme d'une liste. Par exemple : `1, 2, 3, 1, 4, 2, 5, 1`. Vous devez également afficher (sur une nouvelle ligne) le résultat de la fonction `TestCycleEulerien` pour le graphe donné. Par exemple : `True`.

3 Rapport de projet

Vous devez rendre un rapport d'au **maximum 4 pages**. Ce document devra :
- indiquer vos noms et prénoms ;
- proposer une analyse de la complexité en temps et en espace de votre implémentation ;
- indiquer comment exécuter ou compiler votre code (en quelques lignes) ;
- expliquer vos choix d'implémentation, difficultés rencontrées, solutions proposées, ... ;
- si nécessaire, vous pourrez également y présenter le diagramme de classes et un pseudo-code simplifié des méthodes plus complexes.

4 Modalités

Vous réaliserez ce projet par groupe de 2 étudiants. Une soutenance de projet pourra éventuellement avoir lieu après le rendu du projet.

Vous devez rendre une archive (au format *.zip* ou similaire) contenant le code source de votre programme et votre rapport au format PDF. L'archive doit porter le nom des auteurs. Vous devrez déposer cette archive sur la page Celene du cours.

Le projet est à rendre avant le 13 décembre 2024, minuit.

Références

[1] Pierre-Henry Fleury, Deux problèmes de géométrie de situation, *Journal de mathématiques élèmentaires* (1883), 257-261.