
TD 27 : COLORIAGE D'UN GRAPHE

Le fichier `metro_paris_liste_adjacence.py` contient les informations du métro parisien : la variable `Noms` contient la liste des noms des stations du métro dans l'ordre alphabétique et la variable `metro_la` contient la représentation par liste d'adjacence du graphe orienté non pondéré.

Si les fichiers `TD27.py` et `metro_paris_liste_adjacence.py` sont dans le même répertoire, on peut accéder aux variables `Noms` et `metro_la` dans votre fichier de TD par

```
from metro_paris_liste_adjacence import metro_la, Noms
```

Exercice 1 (Coloriage d'un graphe)

Quand on colorie un graphe, on attribue une couleur à chaque sommet (station) de manière à ce que deux sommets adjacents (ou voisins) n'aient jamais la même couleur. Ceci peut être fait par un algorithme **glouton**. Chaque couleur sera représentée par un entier, la première couleur étant associée à l'entier 0.

En coloriant un graphe, on obtient donc un certain nombre de couleur. La question est de savoir si ce nombre de couleur est optimal, c'est à dire le plus petit nombre de couleur nécessaire.

Algorithme glouton de coloriage d'un graphe

Entrée : un graphe orienté non pondéré noté `g`

Sortie : un tableau de couleur (entiers) des sommets

Fonction `COLORIAGE(g)`

`n` ← ordre du graphe

initialiser le tableau couleur de taille `n` à la valeur -1

Pour chaque sommet `s` du graphe

`couleur_voisin` ← liste vide

 Pour tout voisin `v` de `s`

 ajouter la couleur de `v` à la liste `couleur_voisin`

 FinPour

`couleur[s]` ← plus petit entier positif ou nul qui n'appartient pas à `couleur_voisin`

FinPour

retourner le tableau couleur

1. Ecrire la fonction `coloriage` qui implémente l'algorithme de coloriage.
2. Combien de couleurs ont été nécessaires pour colorier le graphe du métro avec l'algorithme glouton si l'on parcourt les sommets dans l'ordre ? Ce résultat est-il optimal ?
3. Afficher dans la console le nom des 3 stations qui nécessitent d'utiliser la cinquième couleur. [Réponse : Châtelet 71, République 314, Stalingrad 341]
4. On décide d'exécuter l'algorithme en parcourant (première boucle `Pour` de l'algorithme) les sommets en sens inverse. Quelles sont les deux stations de la dernière couleur utilisée ?

Exercice 2 (Représentation par matrice d'adjacence du graphe du métro parisien)

On utilise toujours la variable globale `metro_la` (représentation du graphe orienté non pondéré du métro parisien), ainsi que la variable globale `Noms` (noms des stations), importées comme précédemment.

1. Écrire une fonction `la_to_ma` qui prend en comme argument une représentation de graphe orienté non pondéré sous la forme d'une liste d'adjacence (comme par exemple la variable `metro_la`) et qui retourne la matrice d'adjacence représentant le même graphe.
2. Tester votre fonction avec le graphe du métro (éviter cependant d'afficher la matrice en entier!).
3. Écrire une fonction `sens_unique` qui retourne la liste des arcs pour lesquels il n'existe pas d'arc en sens inverse. Vérifier votre fonction avec le graphe du métro.

Réponse : [(`'Danube'`, `'Botzaris'`), (`'Botzaris'`, `'Place des Fêtes'`), (`'Boulogne, Jean Jaurès'`, `'Michel Ange Molitor'`), (`"Porte d'Auteuil"`, `'Boulogne, Jean Jaurès'`), (`'Michel Ange Molitor'`, `'Chardon Lagâche'`), (`'Chardon Lagâche'`, `'Mirabeau'`), (`'Pré-Saint-Gervais'`, `'Danube'`), (`'Mirabeau'`, `'Javel'`), (`'Javel'`, `"Église d'Auteuil"`), (`'Michel Ange Auteuil'`, `"Porte d'Auteuil"`), (`"Église d'Auteuil"`, `'Michel Ange Auteuil'`), (`"Porte d'Auteuil"`, `'Michel Ange Molitor'`), (`'Place des Fêtes'`, `'Pré-Saint-Gervais'`)]

4. Écrire des fonctions `degre_entrant(g,s)` et `degre_sortant(g,s)` qui donnent le degré entrant et le degré sortant d'un sommet `s` du graphe représenté par la matrice d'adjacence `g`. Écrire une fonction `max_degre(g)` qui retourne les noms des stations de degrés maximums.

Trouver les stations qui possèdent les degrés maximums.

Réponse : les 4 stations `Châtelet` et les 5 stations `Républiques` possèdent des degrés entrant et sortant égaux à 6. Il existe une autre station `Châtelet (68)` dont les degrés sont égaux à 5 car c'est le bout de la ligne 11.

5. Il n'existe que deux stations du métro parisien reliées à trois stations dont le nom n'est pas le même que celui de la station. Trouver lesquelles et vérifier sur le plan du métro (on peut faire une recherche dans un fichier pdf; le nom de l'une des deux provient justement de cette propriété).
6. Écrire une fonction `stations_voisines_ma(g, Noms, station)` qui renvoie la liste des noms des stations voisines d'une station `station` dans le graphe `g` (représentation par matrice d'adjacence), dont les noms des sommets sont dans le tableau `Noms`.
7. Améliorer cette fonction pour qu'elle fonctionne si le nom de la station est présent plusieurs fois dans l'ensemble des sommets.

Exemple : `stations_voisines(metro_la, S, 'Opéra')` doit renvoyer [`'Havre Caumartin'`, `'Quatre Septembre'`], [`"Chaussée d'Antin, La Fayette"`, `'Pyramides'`], [`'Madeleine'`, `'Richelieu Drouot'`] car la station `Opéra` est à l'intersection des lignes 3, 7 et 8. On a pris soin de supprimer la station `Opéra` elle-même de la réponse.