PCSI 1 — 2023/2024

Devoir à la maison d'informatique n° 3 :

A rendre pour le : vendredi 17 mai

Les fonctions doivent être commentées.
L'indentation doit être représentée par un trait vertical.
La copie doit être manuscrite.

Coloration de graphes

Dans tout le problème, on considère un graphe G non orienté, connexe et sans boucle. On considèrera que le graphe a pour sommets les éléments de $[\![0, n-1]\!]$, avec $n \in \mathbb{N}^*$.

L'objectif de ce problème est d'attribuer une couleur à chaque sommet du graphe de façon à ce que deux sommets voisins soient de couleurs différentes. On cherchera à minimiser le nombre de couleurs utilisées.

Les couleurs seront représentées par un entier selon le code suivant :

Code couleur

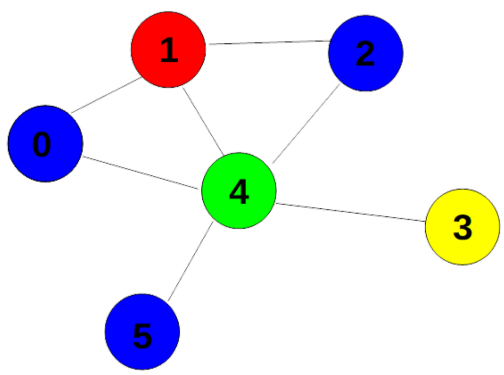
○	0=blanc (absence de couleur)	●	3=jaune
●	1=bleu	●	4=vert
●	2=rouge		....

On considère que le nombre de couleurs est illimité et que le blanc n'est pas considéré comme une couleur mais comme l'absence de couleur.

Un graphe sera représenté par sa matrice d'adjacence donnée sous forme de liste de listes. Sa coloration sera représentée par une liste c de longueur n telle que, pour tout $i \in [\![0, n-1]\!]$, $c[i]$ représente la couleur du sommet i. En particulier, pour un graphe non coloré, on a : $c = [0, \dots, 0]$.

On rappelle que le degré d'un sommet s est le nombre d'arêtes dont s est une extrémité. On appellera liste des degrés la liste d de longueur n telle que, pour tout $i \in [\![0, n-1]\!]$, $d[i]$ est égal au degré du sommet i.

Par exemple, on considère le graphe G_1 suivant :



Sa matrice d'adjacence est représentée par :

$$[[0,1,0,0,1,0],[1,0,1,0,1,0],[0,1,0,0,1,0],[0,0,0,0,1,0],[1,1,1,1,0,1],[0,0,0,0,1,0]].$$

Sa coloration est représentée par :

$$[1,2,1,3,4,1].$$

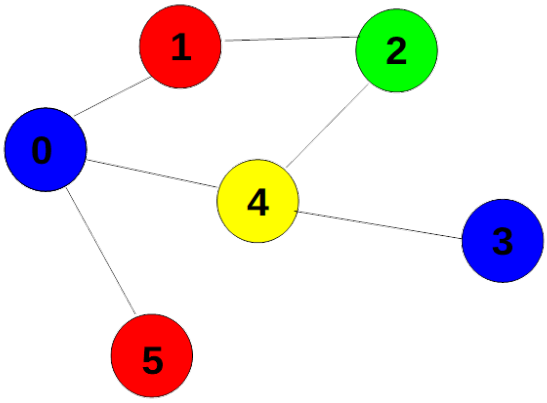
Sa liste des degrés est :

$$[2,3,2,1,5,1].$$

On remarque que deux sommets voisins de G_1 sont toujours de couleurs différentes.

I Questions préliminaires

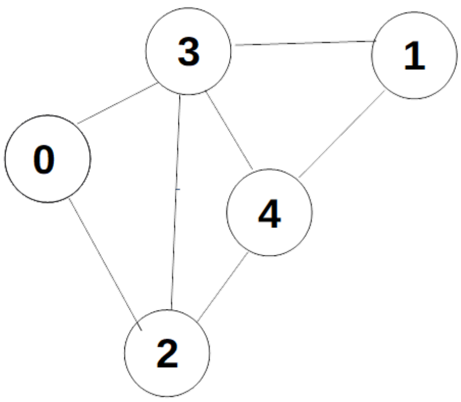
1. On considère le graphe G_2 suivant :



   Donner les représentations de la matrice d'adjacence de G_2, de sa coloration ainsi que la liste de ses degrés.
2. Ecrire une fonction `degre` qui prend comme argument une liste représentant la matrice d'adjacence d'un graphe G et qui renvoie la liste de ses degrés.
3. Ecrire une fonction `voisins` qui prend comme arguments une liste représentant la matrice d'adjacence d'un graphe G ainsi qu'un sommet s et qui renvoie la liste des voisins de s.
4. Ecrire une fonction `PremiereCouleur` qui prend comme arguments une liste représentant la matrice d'adjacence d'un graphe G, un sommet s et une liste de coloration `c` et qui renvoie le code de la première couleur (autre que le blanc) non utilisée par les voisins de `s`.
   Par exemple, pour le graphe coloré G_2 et pour le sommet 4, comme les couleurs utilisées par ses voisins sont 1 et 4, la fonction renvoie 2.

II Algorithme naïf

1. Ecrire une fonction `naif` qui prend comme argument une liste représentant la matrice d'adjacence d'un graphe G et qui renvoie la liste représentant sa coloration selon le principe suivant : on parcourt les éléments s de $[\![0, n-1]\!]$ et la couleur attribuée au sommet s est la première couleur (autre que le blanc) non utilisée par ses voisins.
2. On considère le graphe G_3 suivant :

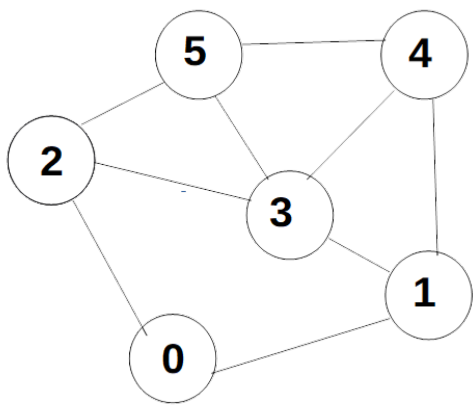


   Quelle est la coloration obtenue par l'algorithme naïf? Le nombre de couleurs utilisées est-il minimal?

III Algorithme glouton

Le principe de l'algorithme glouton est de de parcourir tous les sommets de G en allant du sommet de degré le plus élevé au sommet de degré le moins élevé et d'attribuer à chaque sommet la première couleur (autre que le blanc) non utilisée par ses voisins.

1. Expliquer pourquoi il semble intéressant de commencer par le sommet ayant le degré le plus élevé.
2. En s'inspirant d'une des fonctions classiques de tri, écrire une fonction `TriDegre` qui prend comme argument une liste représentant la matrice d'adjacence d'un graphe G et qui renvoie la liste des sommets de G triée par degrés décroissants.
3. Ecrire une fonction `glouton` qui prend comme argument une liste représentant la matrice d'adjacence d'un graphe G et qui renvoie la liste représentant sa coloration selon l'algorithme glouton décrit ci-dessus.
4. Quelle est la coloration obtenue par l'algorithme glouton pour G_3 ? Le nombre de couleurs utilisées est-il minimal ?
5. On considère le graphe G_4 suivant :



Quelle est la coloration obtenue par l'algorithme glouton ? Le nombre de couleurs utilisées est-il minimal ?

IV Algorithme DSATUR

Soit s un sommet du graphe G muni de couleurs. On appelle degré de saturation du sommet s le nombre de couleurs (autre que le blanc) différentes dans les sommets voisins de s.
Par exemple, pour le graphe G_1, on a :

sommet	0	1	2	3	4	5
degré de saturation	2	2	2	1	3	1

1. Donner les degrés de saturation pour les sommets du graphe G_2.
2. Ecrire une fonction `unique` qui prend comme argument une liste d'entiers `l` et qui renvoie le nombre d'éléments distincts de `l`.
3. Ecrire une fonction `dsat` qui prend comme arguments une liste représentant la matrice d'adjacence d'un graphe G, un sommet `s` et une liste de coloration `c` et qui renvoie le degré de saturation de `s`.
4. Ecrire une fonction `dsatur` qui prend comme argument une liste représentant la matrice d'adjacence d'un graphe G et qui renvoie la liste représentant sa coloration selon l'algorithme suivant :
   - on considère les sommets de G triés par degrés décroissants, on commence par le sommet de plus haut degré,
   - tant que tous les sommets ne sont pas colorés, on choisit un sommet s de G non coloré de degré de saturation maximal, on colore s avec la première couleur non utilisée par ses voisins.
5. Quelle est la coloration obtenue par l'algorithme DSATUR pour G_4 ? Le nombre de couleurs utilisées est-il minimal ?