TD n°8: Graphes

Exercice N°1: Construction de la matrice d'adjacence

Construire une fonction `fMatAdj` qui renvoie la matrice d'adjacence `MatAdj` d'un graphe $\mathcal{G}$. La fonction `fMatAdj` recevra en argument:

- Le nombre de sommets du graphe, ceux-ci étant numérotés de `0` à `n-1`.
- Une liste `L` dont chaque élément sera une liste de deux entiers correspondant à deux sommets adjacents.

Par exemple, avec un graphe de 6 sommets : `L = [[3,5],[1,5],[2,3],[2,1]]`

On note que le 1er et le 5ème sommet n'apparaissent pas dans L: il s'agit de sommets isolés. Avec notre exemple, l'appel à `fMatAdj` sera : `fMatAdj(6,L)`.

Pendant la construction de la matrice d'adjacence, la fonction `fMatAdj` vérifiera la validité de la liste `L` (le cas échéant, elle renverra un message d'erreur du type «Graphe non valide»). En effet, celle-ci ne doit comporter :

- Ni élément de la forme `[i,i]` (pas de boucle);
- Ni doublon : si `[i,j]` est un élément de `L` alors on ne doit pas avoir `[j,i]` dans `L`.

Exercice N°2: Degrés des sommets

On suppose que l'on dispose de l'ordre n d'un graphe $\mathcal{G}$ et de sa matrice d'adjacence `MatAdj`. On demande d'écrire une fonction Python déterminant le degré de chaque sommet. La fonction `fDegSom` recevra comme arguments l'entier n et la matrice `MatAdj`. Elle renverra une liste `D` comportant n éléments, l'élément `D[i]` correspondant simplement au degré du sommet `i`.

Exercice N°3: Coloration

On cherche à attribuer une couleur à chaque sommet d'un graphe $\mathcal{G}$ de telle sorte que deux sommets adjacents <u>ne soient pas</u> de la même couleur.

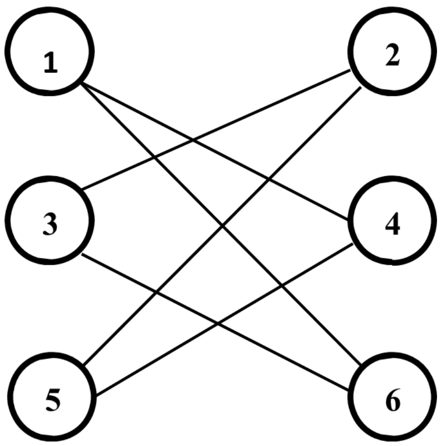
Le nombre de couleurs minimum requis est appelé « **nombre chromatique** » du graphe. On demande d'écrire une fonction Python `welsh_powell` implémentant l'algorithme suivant (algorithme de Welsh et Powell):

1ère étape.
Déterminer les degrés des sommets et les classer dans un ordre décroissant. On devra obtenir une liste DD dont les éléments sont des listes à deux éléments de la forme `[i,d(i)]` où `i` est le numéro du sommet et `d(i)` son degré.

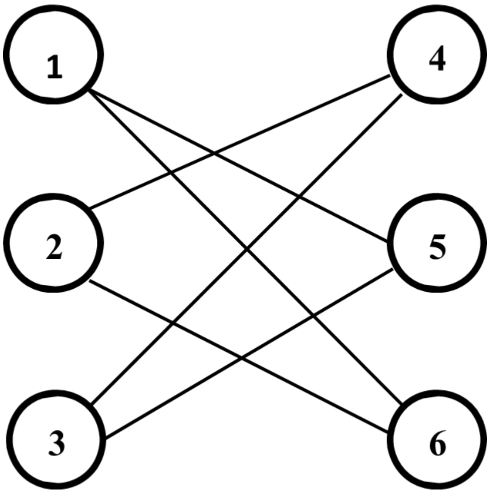
2ème étape.

1) Attribuer au premier sommet de la liste DD une couleur (dans votre fonction, il s'agira de l'entier 0).
2) Attribuer cette même couleur aux sommets de DD non encore coloriés et qui ne sont pas adjacents aux sommets coloriés de cette couleur.
3) Dans la liste DD, considérer alors le premier sommet S non colorié et lui attribuer une nouvelle couleur.
4) Attribuer cette même couleur aux sommets de DD non encore coloriés et qui ne sont pas adjacents aux sommets coloriés de cette couleur.
5) Répéter les étapes 3) et 4) tant que la liste DD contient au moins un sommet non encore colorié.

On testera la fonction sur le graphe suivant :



Supposons maintenant que les sommets du graphe soient numérotés comme suit :



Que renvoie votre fonction ? Que peut-on en conclure quant à l'algorithme de Welsh-Powell ?

Exercice N°4 : Connexité

Soit $\mathcal{G}$ un graphe d'ordre *n* et de matrice d'adjacence A. Expliquer pourquoi la matrice C = $\sum_{k=0}^{n-1} A^k$ permet de savoir si $\mathcal{G}$ est ou non connexe.

Programmer une fonction Python `fGraphConnex` qui reçoit en argument la matrice A et retourne un booléen selon que le graphe $\mathcal{G}$ est ou non connexe.