Coloration de graphe

Comme on l'a vu lors de la précédente séance « *la coloration de graphe consiste à attribuer une couleur à chacun de ses sommets de manière que deux sommets reliés par une arête soient de couleur différente* ».

On cherche à minimiser le nombre de couleurs nécessaires, ce minimum s'appelle nombre chromatique.

Pour colorer un graphe il n'existe pas de méthode efficace qui donne à coup sur le nombre chromatique en un temps raisonnable. Puisque toutes les combinaisons ne peuvent être essayées, certains choix stratégiques doivent être faits c'est ce que l'on appelle une **heurisitique**.

A. Algorithme glouton

Pou chaque entier on associe une couleur et réciproquement et on se donne un graphe $G = (V, E)$.

Pour chaque sommet s

1. On examine l'ensemble $\Omega(s)$ des couleurs déjà attribuées aux voisins de s.
2. On détermine le plus petit entier naturel qui n'appartient pas à $\Omega(s)$.
3. On attribue cette couleur à s.

B. Implémentation en Python

1. On va utiliser une bibliothèque python qui implémente la structure de données de graphe `networkx`. Il faudra l'installer préalablement `pip install networkx`

```
import networkx as nx
from matplotlib import pyplot as plt
G = nx.Graph()
G.add_nodes_from([1,2,3,4,5,6,7,8])
G.add_edges_from([(1,2), (1,3), (1,5), (1,8),
                  (2,3), (3,4), (4,8),(5,6), (6,7), (8,7)])
```

   On va définir un carte de couleurs et visualiser le graphe avec `matplotlib`

```
carte_couleur = [0, 1, 2, 1, 2, 0, 1, 2]
nx.draw(G, node_color = carte_couleur, with_labels=True, font_weight = 'bold')
plt.show()
```

   Tester ce code.

2. Écrire une fonction `coloriage_glouton` qui prend en paramètre un graphe et renvoie un dictionnaire correspondant au coloration du graphe par la méthode gloutonne.

C. Application graphe des spécialités

Dans le dossier moodle récupérer le fichier `T01_07.csv` qui contient les noms anonymisés des élèves de terminale ainsi que leur spécialité.
On souhaite organiser un bac blanc en mai pour cela on doit établir un planning des épreuves. Pour cela on considère un graphe dont les spécialités sont les sommets. Il existe un arête entre deux sommets lorsqu'il existe au moins un élève qui suit les deux spécialités représentées par ces sommets.

1. Dans un premier temps on doit réaliser le graphe entre les différentes spécialités.
   a) On utilise la bibliographystyle `csv` pour exploiter les données du fichier `T01_07.csv` sous la forme d'un dictionnaire.

```
import csv

reader = csv.DictReader(open('T01_07.csv','r',encoding='utf-8'), delimiter=',')
data = [row for row in reader ]
```

   b) Au lycée Bichat il y a 9 spécialités que l'on saisit dans l'odre alphabétique

```
G.add_nodes_from(['HGGSP', 'HLP', 'LLCE', 'LLCA', 'Maths', 'NSI','PH-CH', 'SES', 'SVT'])
```

   Il reste à établir les arêtes entre ces sommets autrement dit déterminer les liens de dépendance entre ces spécialités. On veut de plus un graphe pondéré pour quantifier la relation de dépendance entre deux spécialités. On remarque aussi dans le fichier csv que les binômes de spécialités ne sont pas ordonnées. Par exemple on trouve des élèves avec « Maths, PH-CH » ainsi que des élèves avec « PH-CH, Maths ». Pour palier à ce problème on décide de les ordonnées en utilisant l'ordre alphabétique.
   Compléter et tester le code ci-dessous

```
couple_spe = {}
for dico in data:
    if dico['Spe1'] < dico['Spe2']:
        temp = (dico['Spe1'], dico['Spe2'])
    else:
        temp = (dico['Spe2'], dico['Spe1'])

    if temp  not in couple_spe.keys():
        couple_spe[temp] = .........
    else:
        .......... += 1
```

c) Avec la bibliothèque `networkx` pour ajouter arêtes on utilise les méthodes `add_edges(tab)` ou `add_weighted_edges_from(tab)` ou `tab` est une liste de tuples , respectivement, de la forme `(Sommet1, Sommet2)` ou `(Sommet1, Sommet2, coef)`.

À partir du dictionnaire `couple_spe` définir en compréhension une liste `link` constituée de tuples `(Sommet1, Sommet2, coef)`.

```
link = [..... for ..... in .........]

G.add_weighted_edges_from(link)
```

2. Réprésenter graphiquement ce graphe.

3. Lancer la coloration de ce graphe avec l'heuristique gloutonne puis repésenter graphiquement le graphe coloré. Combien de couleurs sont nécessaires ? Sur combien de séances devront se dérouler les épreuves ?

4. On veut organiser les épréeuves en deux séances en demandant aux enseignants de proposer deux sujets différrents j1 et j2. On s'intéresse maintenant au nombre total N de sujets à écrire.

a) Expliquer pourquoi $N \leqslant 18$.

b) Proposer un planning des examens avec $N < 18$.

c) Déterminer la valeur minimale possible pour N dans le cas de notre établissement.