

P16

THÉORIE DES GRAPHES I

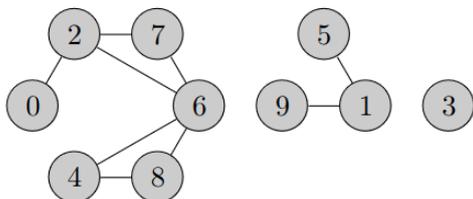
Représenter un graphe en python

Le but de ce TP est de représenter en python des graphes et de reprendre les notions vues en cours de mathématiques.

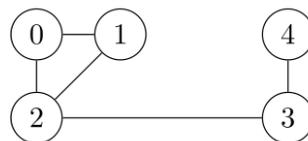
I Matrice d'adjacence

On considère les graphes non orientés suivants (G_1 et G_2).

G_1 :



G_2 :



Exercice 1

1. Définir leurs matrices d'adjacence A_1 et A_2 dans le format `np.array`.

2. Écrire une fonction qui renvoie la liste des degrés de chaque sommet d'un graphe donné par une matrice A . On pourra remarquer que le degré du sommet i est la somme des coefficients de la ligne i .

3. Écrire une fonction qui renvoie le nombre de chemins de longueur k entre les sommets i et j . On utilisera la fonction `al.matrix_power`.

4. Écrire une fonction qui renvoie `True` si tous les coefficients d'une matrice carrée M sont strictement positifs et `False` sinon.

5. En déduire, à l'aide de la propriété du cours, une fonction qui renvoie `True` si le graphe de matrice A est connexe, et `False` sinon.

6. Écrire une fonction `voisins(A,i)` qui, étant donné un sommet i dans un graphe de matrice A , renvoie la liste des voisins de ce sommets (les sommets qui lui sont adjacents).

II Liste d'adjacence

Le problème de la matrice d'adjacence est qu'elle contient beaucoup de données inutiles (on doit souvent saisir un grand nombre de 0) et devient donc rapidement peu pratique.

On va voir une autre façon de représenter les graphes : les listes d'adjacence.

On donne une liste qui contient, pour chaque sommet, la liste de ses voisins.

Exercice 2 1. On a représenté G_1 par une liste d'adjacence. Faire de même pour G_2 .

```
G1 = [[2], [5,9],[0,6,7], [], [6,8], [1], [2,4,7,8], [2,6], [4,6],  
      [1]]
```

2. Vérifier que votre fonction voisins de l'exercice 1 permet de retrouver cette liste d'adjacence.

3. Réécrire une fonction renvoyant le degré des sommets, en utilisant les listes d'adjacence.