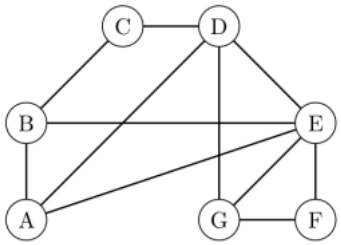
Python 9 : Graphes

1. La matrice d'adjacence

Un graphe peut se modéliser en Python par sa matrice d'adjacence.
Par exemple, le graphe suivant :



Peut se modéliser par sa matrice d'adjacence avec la commande :
M=np.array([*remplir avec les coefficients de la matrice d'adjacence* …])
C'est très fastidieux, car il y a n^2 coefficients à remplir (n est l'ordre du graphe).

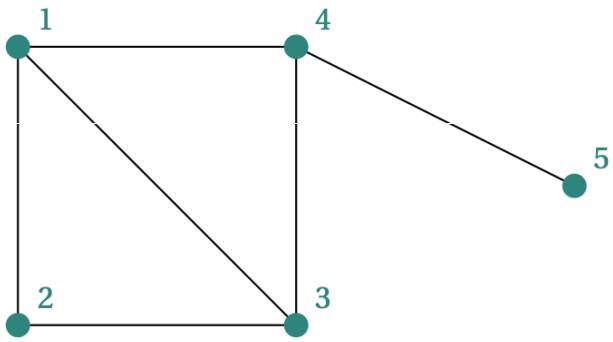
Disposer de la matrice, ainsi que de la bibliothèque numpy.linalg permet de calculer les puissances de la matrice et ainsi de récupérer le nombre de chemins d'une longueur donnée reliant les paires de sommets entre elles.

Quelques remarques :
Étant donné un graphe simple, le degré du sommet i est égal au nombre de 1 présents sur la colonne i (ou la ligne i).
Un sommet isolé se repère donc avec une colonne (ou une ligne) de zéro.

2. La liste des listes d'adjacence

Étant donné une liste de sommets S=[s1,…sn] on peut associer à chaque sommet s_i une liste composée des sommets adjacents au sommet s_i. On regroupe ces listes dans une liste (on a donc une liste composée de n éléments dont chaque élément est une liste).

Prenons un exemple :



La liste des listes d'adjacence est :
L=[[2,3,4] ; [1,3] ; [2,4] ; [1,3,5] ; [4]]

Quelques remarques :
Si un sommet est isolé sa liste d'adjacence est [].
Dans un graphe simple la longueur de la liste d'adjacence du sommet i est égale au degré du sommet i.
La matrice d'adjacence et la liste des listes d'adjacences représentent la même information de deux manières différentes. On préférera la matrice lorsqu'on veut calculer les puissances.