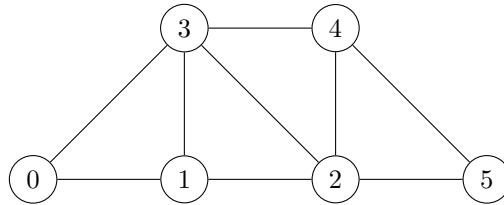


TP D'INFORMATIQUE 12

Représentation des graphes

L'objectif de ce TP est de manipuler les différentes façon de représenter un graphe. On se placera dans le cas de graphes non orientés, non pondérés ni étiquetés, à l'exemple du graphe G suivant :



1 Opérations fondamentales sur les graphes

1. Définir en Python la liste `G1` représentant le graphe G par liste de listes d'adjacence.
2. Définir en Python la liste `M1` représentant le graphe G par une matrice d'adjacence.
3. Écrire une fonction `degre1` prenant en entrée un graphe G représenté par une liste des listes d'adjacence et un sommet u , et renvoyant le degré de u .
4. Écrire une fonction `degre2` prenant en entrée un graphe M représenté par une matrice d'adjacence et un sommet u , et renvoyant le degré de u .
5. Sur le même modèle, pour chacune des fonctions suivantes, écrire une version travaillant sur une liste de listes d'adjacences, et une version travaillant sur une matrice d'adjacence :
 - (a) Une fonction `ordre` prenant en entrée un graphe, et renvoyant son ordre, c'est-à-dire son nombre de sommets.
 - (b) Une fonction `sont_voisins` prenant en entrée un graphe ainsi que deux sommets u et v , et testant si ces sommets sont voisins dans le graphe.
 - (c) une procédure `ajoute_sommet` prenant en entrée un graphe et ajoutant un nouveau sommet isolé.
 - (d) Une procédure `ajoute_arete` prenant en entrée un graphe et une arête $a = [u, v]$ et ajoute l'arête au graphe. Les sommets u et v sont supposés appartenir au graphe, et que l'arête n'existe pas déjà. On ne cherche pas à conserver triées les listes d'adjacences.
 - (e) Une fonction `nb_aretes` prenant en entrée un graphe et renvoyant son nombre d'arêtes.
 - (f) Une fonction `regulier` prenant en entrée un graphe et testant s'il est régulier, c'est-à-dire si tous ses sommets sont de même degré.
 - (g) Une fonction `aretes` prenant en entrée un graphe, et renvoyant la liste de ses arêtes, sous forme de listes de couples (chaque arête ne doit apparaître qu'une seule fois).
6. Déterminer la complexité des fonctions précédentes, en fonction du nombre de sommets n du graphe.
7. Écrire la fonction `liste_vers_matrice` prenant en entrée un graphe représenté par listes d'adjacence et renvoyant la matrice d'adjacence correspondante.
8. Écrire la fonction inverse `matrice_vers_liste` qui prenant en entrée une matrice d'adjacence et renvoyant la liste des listes d'adjacence correspondante.

2 Puissances de la matrice d'adjacence

La représentation par matrice d'adjacence offre l'avantage de pouvoir utiliser les outils de l'algèbre matricielle pour analyser le graphe.

1. Écrire une fonction `produit` qui calcule le produit de deux matrices carrées de même taille, représentées par des listes de listes.

2. En déduire une fonction **puissance** calculant la puissance k -ième d'une matrice carrée.
3. En étudiant différents exemples, conjecturer à quoi correspond le coefficient de coordonnées i, j de la puissance k -ième de la matrice d'adjacence d'un graphe donné.
4. Un **triangle** d'un graphe non orienté est un ensemble de trois sommets voisins deux à deux. On cherche dans cette question à calculer le nombre de triangles dans un graphe (par exemple le nombre de trios d'amis dans un réseau social).
 - (a) La **trace** $\text{tr}(M)$ d'une matrice carrée M est la somme de ses coefficients diagonaux. Justifier que le nombre de triangles dans un graphe de matrice d'adjacence M est donné par $\text{tr}(M^3)/6$.
 - (b) Écrire une fonction **nb_triangles** prenant en entrée une matrice d'adjacence et renvoyant le nombre de triangles du graphe représenté.
5. Écrire une fonction **non_voisins_les_plus_proches** prenant en argument une matrice d'adjacence, et calculant les deux sommets non voisins ayant le plus de voisins en commun.
6. La centralité de Katz est une mesure de la prépondérance d'un sommet dans la structure des chemins d'un graphe, qu'on peut par exemple interpréter comme l'influence d'un individu dans un réseau social. La centralité de katz d'un sommet u est donnée par la formule :

$$C_K(u) = \sum_{k=1}^{+\infty} \sum_{v=0}^{n-1} \alpha^k M_{v,u}^k$$

où $\alpha < 1$ est le **facteur d'atténuation**.

Écrire une fonction **katz** prenant en entrée M , u et α , et renvoyant une approximation de $C_K(u)$. On stoppera le calcul de la somme infinie à la première valeur de k qui ne change pas sa valeur de plus de 10^{-9} . On prendra garde à ne pas refaire k produit matriciels pour chaque valeur de k .

7. Écrire une version des fonctions de cette partie prenant en argument un réseau social donné sous forme de dictionnaire de listes d'adjacences (dans lequel chaque sommet est une chaîne de caractères).