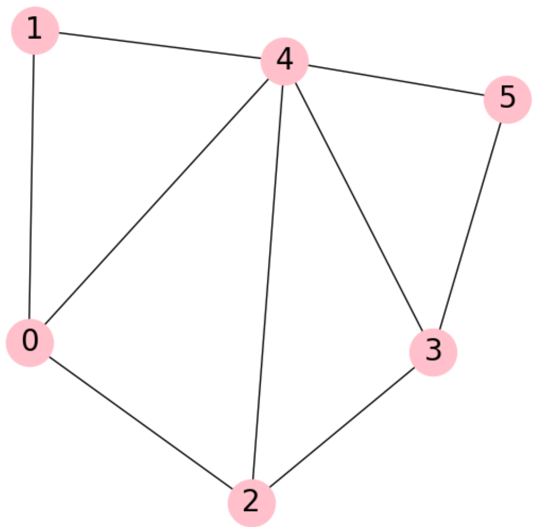
TP10 : Graphes (partie 1)

Implémenter un graphe non orienté sous la forme d'un dictionnaire

1. On considère le graphe $\mathscr{G}_1$ non orienté suivant.
   Sur un code à récupérer sur ma page web (et à enregistrer dans un script nommé `TP10.py`), on a implémenté $\mathscr{G}_1$ sous la forme d'un dictionnaire où chaque clé correspond à un sommet auquel on associe sa liste de sommets voisins.



2. **Retirer un sommet d'un graphe**.
   (a) Écrire une fonction `retraitSommet` prenant en arguments un graphe `G` (codé sous la forme d'un dictionnaire) et le nom d'un sommet `s` du graphe et qui renvoie le graphe obtenu en retirant ce sommet (et ses arêtes adjacentes) du graphe.
   On fera en sorte que cette fonction ne soit pas à effet de bord : elle ne doit pas modifier le graphe passé en argument.
   On pourra créer (et travailler avec) une copie indépendante du dictionnaire codant le graphe, utiliser l'instruction `del[s]` pour éliminer le sommet d'indice `s` du graphe, puis enlever dans les voisins de chaque sommet restant le sommet `s` à l'aide de la méthode `remove` sur les listes : par exemple, si L est une liste, alors `L.remove(s)` supprime le premier (ici le seul) `s` en partant de la gauche.
   (b) Tester cette fonction en retirant le sommet 0 du graphe $\mathscr{G}_1$. Représenter ci-dessous le graphe $\mathscr{G}_2$ ainsi obtenu.

3. **Degré d'un sommet**.
   (a) Que doit-on taper dans le Shell pour obtenir le degré du sommet 0 ? ....

(b) Écrire une fonction `sommetMaxDegre` prenant en argument un graphe implémenté sous la forme d'un dictionnaire et qui renvoie le nom du ou des sommets de plus haut degré du graphe.

(c) Quel est le sommet de plus haut degré du graphe $\mathscr{G}_1$? ....

Matrice d'adjacence d'un graphe

4. (a) Écrire une fonction `dicToMat` qui prend en argument un graphe `G` implémenté sous la forme d'un dictionnaire et qui renvoie la matrice d'adjacence correspondante (on ordonnera les sommets par ordre croissant ou alphabétique à l'aide la méthode `L.sort()` permettant de trier une liste `L` en place).

   (b) Tester cette fonction avec le graphe $\mathscr{G}_1$ en argument (dont la matrice d'adjacence a déjà été déterminée à la main dans la fiche de cours).

5. (a) On considère un graphe $\mathscr{G}_3$ dont les sommets sont numérotés de 0 à 3. Sa matrice d'adjacence s'écrit

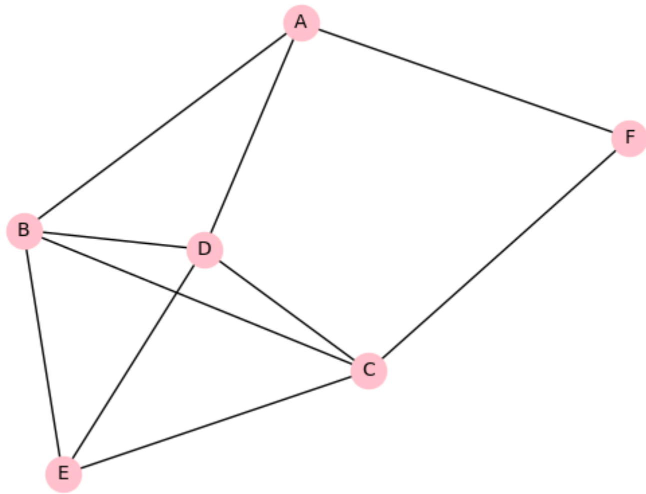
$$\begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

   Dessiner le graphe correspondant.

   (b) Combien y a-t-il de chemins de longueur 3 reliant les sommets 0 et 3 ? ....
   On pourra utiliser la fonction `dot` *du module* `numpy` *pour obtenir rapidement cette information.*

   (c) Écrire une fonction `MatToDic` qui prend en argument une matrice carrée `M` de taille n et qui renvoie un graphe correspondant implémenté sous la forme d'un dictionnaire où les sommets seront numérotés de 0 à $n-1$.

Une fourmi est au point A sur le graphe $\mathscr{F}$ ci-contre. Elle se déplace sur le graphe de sommet en sommet en choisissant au hasard une arête (y compris celle qu'elle vient d'emprunter) quand elle est sur un sommet.



6. Implémenter le graphe $\mathscr{F}$ sous la forme d'un dictionnaire.
   On pourra, au choix, conserver les lettres pour désigner les sommets ou numéroter les sommets.

7. **Probabilité de présence sur un sommet**.
   (a) Écrire une fonction `arrivee` prenant en argument un entier naturel `n` et qui renvoie la position de la fourmi sur le graphe $\mathscr{F}$ après `n` déplacements.

   (b) Écrire une fonction `frequence` prenant en argument deux entiers naturels `n` et `nbRep` et le nom `sommet` d'un sommet et qui renvoie la fréquence à laquelle la position de la fourmi après `n` déplacements est le sommet `sommet` lors de `nbRep` marches aléatoires différentes à partir du sommet A sur le graphe $\mathscr{F}$.
   On pourra importer le module `random` avec l'alias `rd` et utiliser la fonction `rd.choice(L)` permettant d'obtenir de façon aléatoire et équiprobable un élément d'une liste `L`.
   (c) Estimer la probabilité que la fourmi soit au sommet C après 10 déplacements.

   ....

   (d) Déterminer la matrice d'adjacence associée au graphe $\mathscr{F}$.

   (e) Déterminer le nombre de chemins de longueur 10 reliant les sommets A et C. ....

   (f) Déterminer le nombre total de chemins de longueur 10 partant du point A. ....

   (g) Comment pourrait-on alors obtenir la probabilité que la fourmi soit au sommet C après 10 déplacements ? Comparer avec la valeur estimée précédemment. Commenter.

   ....

   ....