# TP 8 – Graphes et matrices d'adjacence

L'objet de ce TP est d'illustrer de manière concrète les notions vues dans le cours sur les graphes.

Dans tout le TP, les matrices seront manipulées sous la forme de tableaux numpy (type `array`).

**Questions :**

1. Dessiner le graphe dont la matrice d'adjacence est $\begin{pmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$.

   Quelle est la liste d'adjacence associée ?
2. Écrire un programme prenant en entrée la matrice d'adjacence d'un graphe et le numéro d'un sommet du graphe, et retourne le degré entrant de ce sommet.
3. On souhaite trouver la liste des sommets atteignables à partir d'un sommet donné.
   a. Quel résultat du cours permet de déterminer s'il existe un chemin élémentaire de longueur $k$ allant d'un sommet $s_1$ à un sommet $s_2$ ?
   La proposition 3.4
   b. Écrire un programme qui prend en entrée la matrice d'adjacence d'un graphe $G$ et les numéros de deux sommets $s_1$ et $s_2$, et dit s'il existe ou non un chemin de longueur $k$ allant de $s_1$ vers $s_2$.
   *Indication :* *on pourra utiliser la fonction* `matrix_power(A,k)` *du module* `linalg` *de la bibliothèque* `numpy`.
   c. Si un graphe possède $n$ sommets, quelle est la longueur maximale d'un chemin qui ne passe pas deux fois par le même sommet (sauf éventuellement ses extrémités qui peuvent être égales) ?
   $n-1$
   d. En déduire un programme qui prend en entrée la matrice d'adjacence d'un graphe $G$ et les numéros de deux sommets $s_1$ et $s_2$, et détermine s'il existe ou non un chemin allant de $s_1$ vers $s_2$.
   e. Modifier le programme de façon à ce qu'il retourne la distance de $s_1$ à $s_2$, c'est-à-dire la plus petite longueur d'un chemin allant de $s_1$ vers $s_2$.
4. On souhaite modifier le programme de la question précédente de façon à étudier les circuits dans un graphe.
   a. Écrire un programme prenant en entrée la matrice d'un un graphe $G$ et retournant le nombre de boucles dans $G$.
   b. Quel résultat du cours permet de trouver le nombre de circuits de longueur $k$ dans un graphe ?
   Cas particulier de la proposition 3.4 .
   c. Écrire un programme prenant en entrée la matrice d'un un graphe $G$ et un entier $k \geqslant 1$, et qui retourne le nombre de circuits de longueur k dans $G$.
   d. En déduire un programme prenant en entrée la matrice d'un un graphe $G$ et qui décide si $G$ est acyclique.

**5.** Tracer le graphe dont la liste d'adjacence est

`[[2,7],[2,3,4],[6],[2,4],[2,5],[6,7],[],[2]]`.

**6.** Écrire une fonction `MatriceAdjacence(L)` qui prend en entrée la liste d'adjacence d'un graphe $G$ et retourne la matrice d'adjacence de $G$

**7.** Écrire une fonction `ListeAdjacence(M)` qui prend en entrée la matrice d'adjacence d'un graphe $G$ et retourne la liste d'adjacence de $G$.

**8.** Écrire et documenter une fonction miroir qui prend en argument la liste d'adjacence d'un graphe $G$ et retourne la liste d'adjacence du graphe obtenu en retournant les arcs de $G$.

$\sim$