

## Théorie des Graphes

### TP 01

#### Exercice 1

Ecrire un programme qui contient :

- L'initialisation d'un graphe d'ordre  $n$  en utilisant
  - Une matrice d'adjacence (Python : liste 2D)
  - Une liste d'adjacence (Python : Dictionnaire)
- Une fonction **chemin** ayant comme paramètres :
  - un graphe orienté  $G$
  - un sommet **origine**
  - un sommet **extremite**
  - une longueur  $m$

La fonction doit afficher tous les chemins possibles entre les sommets **origine** et **extremite** de longueur  $m$

Indication : si vous utilisez le produit matriciel, vous pouvez utiliser l'exponentielle rapide (En : square-and-multiply).

- Affiche tous les circuits de longueur  $\leq n$

#### Exercice 2 : Composantes fortement connexes

Ecrire un programme qui initialise un graphe orienté  $G$  d'ordre  $n$  et détermine ses composantes fortement connexes en utilisant la méthode de Demoucron.

#### Exercice 3 : Coloration de graphe

Ecrire un programme qui initialise un graphe non orienté  $G$  d'ordre  $n$  et le colorie en utilisant l'algorithme de Welsh & Powell

#### Exercice 4 : Arbre couvrant minimal

- Ecrire une fonction qui vérifie l'existence d'un circuit dans un graphe  $G$
- Ecrire une fonction qui implémente l'algorithme de Kruskal vu au cours
- Ecrire une fonction qui implémente l'algorithme de Prim vu au cours
- Ecrire un programme qui initialise un graphe  $G$  non orienté, connexe, **valué**, d'ordre  $n$  et affiche un arbre couvrant de poids minimal.