

## DEVOIR LIBRE

**A RENDRE SUR MOODLE :** Vos programmes. Puis un 'makefile' ou un 'readme' de comment compiler et/ou exécuter vos programmes. Un petit 'rapport' de maximum 5 pages.

**Travail individuel.**

A rendre sur Moodle le 28 Février 2026. Compresser le tout et mettre dans un fichier Nom.tar ou Nom.zip ou Nom.tar.gz

**Exercice 1 : Des cliques**

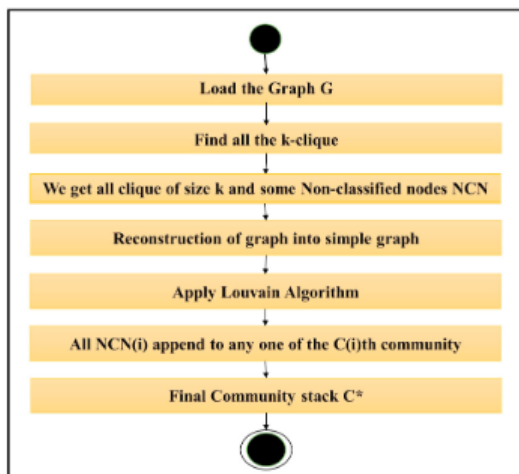
Coder pour générer le graphe suivant ( $n$ ,  $p$ ,  $q$  sont des paramètres qui seront amener à fluctuer) :

- Le graphe a  $n$  sommets réparti au uniformément aléatoirement au hasard (uah) dans 4 lots de sommets :  $n_1, n_2, n_3, n_4$  (tels que  $\sum_{i=1}^4 n_i = n$ ).
- Dans chaque lot de sommets, les sommets entre eux sont connectés avec une probabilité  $p$ .
- Pour deux sommets dans deux différents lots, la probabilité de connexion est  $q$ .
- Dessiner le graphe obtenu.

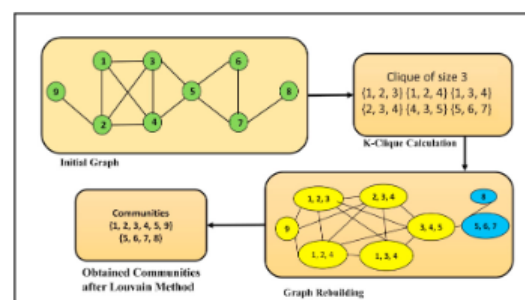
1. En utilisant la fonctionnalité 'k-clique-communities' et selon les valeurs de  $p$  et  $q$ , calculer la distribution des  $k$ -cliques dans votre graphe. Commentez vos résultats.

**Exercice 2 : Modifications de Louvain**

Nous avons vu qu'il y a plusieurs algorithmes possibles ('Louvain' est une possibilité parmi d'autres) pour détecter des communautés. Dans cet exercice, nous allons modifier l'algorithme de Louvain selon le plan :



(a) Shows the various steps flow chart of the CBLA algorithm.



(b) Show step by step process of CBLA method with the help of an example.

1. Comparer les résultats entre 'Louvain' et les modifications faites selon les idées décrites par la figure ci-dessus.
2. Faire un tableau comparatif (temps d'exécution, passage à l'échelle, ...) des algorithmes, selon les valeurs de  $n$ ,  $p$ ,  $q$  de l'exercice précédent. Commentez.