

# Détection de communautés dans des réseaux sociaux

## Introduction :

Le but du projet est de vous initier à la lecture et la compréhension d'articles scientifiques. Dans ce but, nous demandons de lire des articles récents et d'implémenter certains algorithmes de graphes. Le contexte est celui des réseaux sociaux et de leur étude. Un réseau social peut facilement être vu comme un graphe où les nœuds représentent les utilisateurs et les arêtes les liens qui existent entre eux dans le réseau. Par exemple pour Facebook, le lien peut représenter un lien d'amitié. La problématique principale du projet est d'énumérer certaines structures dans des graphes modélisant théoriquement certains réseaux sociaux. En particulier on va chercher dans ces graphes des communautés, c'est-à-dire des ensembles de nœuds très densément connectés entre eux. Connaître la structure des communautés dans un graphe trouve de nombreuses applications. Cela permet, par exemple, de simplifier le ciblage publicitaire. En théorie des graphes, on modélise souvent les communautés par des cliques maximales dont la définition est donnée plus tard dans le projet.

## Instructions :

Le but du projet est d'implémenter deux algorithmes d'énumération de cliques maximales dans des graphes générés aléatoirement. Le projet s'articule en trois parties.

**La première partie** consiste à générer des graphes aléatoires de deux façons différentes. Premièrement, étant donné un nombre fixe de sommet, chaque arête apparaît avec probabilité  $p$  choisie aléatoirement telle que  $0 < p < 1$ .

La seconde famille de graphes aléatoires que l'on vous demande de générer sont les graphes de Barabási-Albert. Ces graphes sont construits itérativement de la façon suivante. On considère un paramètre  $m$  (ici on prendra  $m=2$ ). Initialement le graphe est un graphe triangle (une clique à trois nœuds). On ajoute des nœuds un par un. Supposons que les nœuds du graphe existant sont numérotés. Un nouveau nœud sera connecté au nœud  $i$  avec probabilité  $\text{degré}(i)/\sum \text{degrés}$ . On créera au plus  $m$  nouvelles arêtes. Par exemple, lorsque l'on ajoutera le quatrième nœud au triangle initial, il sera connecté avec probabilité  $2/6$  à chacun des nœuds du graphe existant.

**La seconde partie** du projet consiste à écrire un algorithme de complexité exponentielle pour l'énumération des cliques. Soit un graphe  $G=(V,E)$  avec  $V$  l'ensemble des sommets et  $E$  l'ensemble des

arêtes de  $G$ . Une clique de  $G$  est un ensemble  $K$  de sommets du graphe tous connecté deux à deux. La clique  $K$  est maximale si elle n'est incluse dans aucune autre clique de  $G$ , c'est-à-dire qu'il n'existe pas de sommet dans  $V \setminus K$  connecté à tous les sommets de  $K$ . L'algorithme de Bron Kerbosch est un des algorithmes les plus connus énumérant les cliques maximales d'un graphe. Il en existe de nombreuses variantes. On vous demande d'implémenter la version standard et une version améliorée qui considère les sommets dans un ordre calculé à l'avance qui s'appelle l'ordre de dégénérescence du graphe. Vous trouverez cet algorithme dans l'article scientifique [1]. Vous aurez donc aussi besoin dans un premier temps de calculer l'ordre de dégénérescence du graphe (l'algorithme est décrit dans le même papier).

Finalement, dans **la troisième partie**, on vous demande d'implémenter les deux algorithmes d'énumération des cliques maximales présentés dans [2]. Vous êtes libre pour le choix de vos structures de données, hormis celles concernant le graphe qui seront stockés sous forme de listes d'adjacence. Vous pourrez aussi changer librement la façon dont sont stockés certains résultats dans l'algorithme (par exemple un arbre des suffixes est utilisé dans le papier; vous pouvez le remplacer par une table de hachage etc.). Vous utiliserez les algorithmes implémentés dans la seconde partie du projet pour la sous routine d'énumération des cliques maximales.

## **Rendu :**

Un rapport est attendu, dans lequel on veut une présentation théorique des algorithmes et une comparaison sur différentes instances des graphes générés en première partie de projet en terme de temps et d'espace d'exécution (en fonction du nombre de sommets et d'arêtes). On attend aussi à ce que vous décriviez précisément le contenu de ce que vous avez fait. Le langage de programmation sera du C++. On demande aussi à ce que le code soit propre, commenté et fournit avec un ReadMe décrivant comment le compiler et l'exécuter, ainsi qu'un exemple d'exécution des algorithmes sur des graphes générés en première partie de projet.

## **Références :**

[1] Eppstein, Löffler, Strash, *"Listing All Maximal Cliques in Sparse Graphs in Near-optimal Time"*, ISAAC 2010.

[2] Manoussakis, « *A new decomposition technique for maximal clique enumeration for sparse graphs* », Theoretical Computer Science 2019.