Université Polytechnique Hauts-de-France
Institut National des Sciences Appliquées Hauts-de-France **2020-2021**

Spécialité Informatique par Apprentissage 1ère année Graphes et Algorithmes

Fiche d'exercices pratiques 3

On considère ici uniquement des graphes non orientés. Vous pouvez utiliser une des deux structures de données, au choix.

Exercice 1 :

Dans cet exercice on cherche à implémenter un algorithme permettant de trouver des cliques maximales dans un graphe non orienté $G = (S, A)$. Pour rappel une clique est un ensemble de K sommets deux à deux adjacents. Une clique correspond ainsi à un sous-graphe complet de G. La taille d'une clique est égale au nombre de sommets qui la composent. Une clique est dite **maximale** s'il n'existe pas une autre clique plus grosse la contenant. Il ne faut donc pas confondre clique maximale et clique maximum, ou de taille maximale. Une méthode pour trouver une clique maximale peut être formulée comme suit :

(i) Démarquer tous les sommets de S et toutes les arêtes de A ;

(ii) Marquer un sommet du graphe ;

(iii) Tant qu'il existe un sommet s non marqué et tel que tout sommet marqué de S admet s comme voisin, marquer ce sommet s ;

(iv) Marquer toutes les arêtes reliant deux sommets marqués. L'ensemble des sommets marqués forment une clique maximale.

Cet méthode peut être répétée, à partir de l'étape (ii), tant qu'il reste des sommets non marqués dans le graphe, afin de construire plusieurs cliques maximales.

1. Implémenter une version de cette approche.

Afin d'éviter de toujours générer les mêmes cliques, et ainsi augmenter la probabilité de trouver des cliques de plus grandes tailles, on peut introduire de l'aléatoire dans la méthode précédente.

2. Écrire une fonction permettant de récupérer un sommet non marqué, choisi aléatoirement.
3. Écrire une fonction qui retourne un sommet choisi aléatoirement parmi l'ensemble des sommets non marqué vérifiant les conditions de l'étape (iii) de la méthode précédente, s'il en existe un.
4. Écrire une fonction exploitant ces deux fonctions pour donner une version aléatoire de la méthode précédente.
5. Comparer les deux méthodes sur différents graphes, et en exécutant plusieurs fois la méthode aléatoire, afin de vérifier expérimentalement si l'aléatoire permet d'améliorer le comportement de la méthode initiale.