Institut National des Sciences Appliquées **2019 - 2020**

Licence 3 Informatique Graphes et Algorithmes

T.P. 3 : Parcours du graphe et connexité

Modalités pour chaque séance de TP :

- Les programmes seront développés de préférence **en langage C**. L'utilisation d'un autre langage ne sera acceptée que si la mémoire **dynamique** est gérée par l'utilisateur (allocation et libération explicites dans le code).
- Vos programmes doivent pouvoir être compilés et exécutés sous Windows **et** Linux.
- **Attention :** ne seront récupérés que les fichiers sources (.h et .c en 'C'). Par conséquent, les codes sources pour lesquels un **Makefile** (pour Linux) et un **ReadMe** auront été fournis seront **les seuls évalués**. Les archives contenant des *fichiers projets* générés à partir d'un éditeur quelconque (Eclipse, Code Blocks, NetBeans, Visual Studio, ...) **ne seront pas acceptées**. Vous devez fournir les informations nécessaires à l'enseignant pour qu'il puisse générer l'exécutable sans avoir à utiliser un logiciel particulier.

Évaluation (lire attentivement)

Votre travail sera évalué de la façon suivante :

- **1 note sur 10 pour le travail réalisé durant les 2 dernières séances ($\Rightarrow$ 1 note sur 20).**
- Le TP2 et TP3 doivent obligatoirement être déposés sur moodle avant la fin de la séance **par chaque étudiant** (même si vous travaillez en groupe). Les traveaux envoyés par mail ne seront pas acceptés.
- **Si votre code ne s'exécute pas et si vous n'avez mis aucun commentaire dans le code, ceci entraînera un 0.**
- Vous pouvez travailler en groupe de **max 2 étudiants**. Mettre en commentaire dans chaque fichiers les noms de tous les étudiants du groupe.

Une absence à un TP entraînera un 0, sauf si cette absence est justifiée (certificat médical ou raison **validée par l'enseignant**).

But du TP3 :

Vous allez maintenant appliquer les fonctions de la bibliothèque `graphe_matrice` et `graphe_liste` pour resoudre des problèmes sur les graphes. Chaque excercice doit être réalisé en utilisant les deux structures de données : listes et matrice.

Pour simplifier la manipulation et le développement, on suppose **dans toute la suite** que les graphes disposent de n sommets numérotés de 0 à n-1. De plus le graphe sera noté avec $G = (S, A)$, où S représente l'ensemble des sommets et A l'ensemble des arcs et $n = |S|,\ m = |A|$.

1 Parcours en profondeur

Réaliser cet exercice uniquement avec des listes.

Pour un graphe non-orienté G donné (lecture de fichier ou génération aléatoire) et le somment k , $0 \leq k \leq n-1$, appliquer un parcours en profondeur à partir du sommet k.

2 Parcours en largeur

Réaliser cet exercice uniquement avec des listes.

Pour un graphe orienté G donné (lecture de fichier ou génération aléatoire) et le somment k, $0 \leq k \leq n-1$, appliquer un parcours en largeur à partir du sommet k.

3 Connexité

Pour un graphe non-orienté G donné (lecture de fichier ou génération aléatoire) trouver le plus grand graphe réduit. Quelle est la complexité de votre algorithme ?