

Travaux Pratiques

Graphes

Jean-Christophe Deneuille
<jean-christophe.deneuille@xlim.fr>

Année Universitaire 2015 - 2016

Dans ce projet, vous allez construire en binôme une bibliothèque de manipulation de graphes en C. Vous disposerez de 5 séances encadrées pour réaliser ce travail :

- jeudi 28 avril, 14h00 - 15h30
- jeudi 28 avril, 15h45 - 17h15
- vendredi 29 avril, 14h00 - 15h30
- vendredi 29 avril, 15h45 - 17h15
- mardi 10 mai, 9h45 - 11h15

À l'issue de ces 5 séances vous devrez remettre un rapport synthétisant votre travail. Les consignes pour le rapport sont les suivantes :

- 6 pages minimum,
- 10 pages maximum,
- Rédigé en \LaTeX ,
- Références clairement identifiées,
- Description de l'implication de chaque membre dans le projet,
- Deadline : dimanche 15 mai, 23h59 heure française.

Vous compresserez votre rapport (.pdf), ses sources (.tex), ainsi que votre code (.c et .h) en une archive (.zip ou .tar.gz) que vous hébergerez sur votre compte **FileX** (votre archive devra être téléchargeable jusqu'à une semaine après la deadline). Vous devrez m'envoyer le lien de téléchargement par mail (adresse en entête de sujet) à partir de votre adresse **@etu.unilim.fr** avant la deadline. Vous recevrez un accusé de réception dans les 24 heures suivant l'envoi de votre mail. En cas de non-réception de cet accusé, il vous appartient de me contacter dans les plus brefs délais.

Tout retard sera sanctionné à hauteur d'un point par tranche de 6 heures (heure du serveur SMTP d'unilim faisant foi). Le non-respect de ces consignes pourra également entraîner une perte de points. Le cours est disponible [ici](#).

Exercice 1 Séances 1 et 2

L'objectif de ces premières séances est d'implémenter les structures et méthodes/fonctions de base pour la manipulation de graphes :

1. Utilisation d'une structure de données pour stocker un graphe en mémoire centrale (Matrice d'adjacence, d'incidence, ...),
2. Chargement/Sauvegarde d'un graphe depuis/vers un fichier texte,
3. Algorithmes de parcours d'un graphe en largeur et en profondeur,
4. Mise en œuvre des méthodes de calcul de composantes connexes et fortement connexes,
5. Test des fonctionnalités de base sur des exemples simples.

Exercice 2 Séances 3 et 4

Dans cette seconde partie, vous choisirez 3 algorithmes parmi les suivants et vous les implémenterez :

1. Kruskal
2. Prim
3. Dijkstra
4. Bellman
5. Welsh-Powell (non vu en cours → curieux ?)

Vous testerez vos algorithmes sur des exemples simples.



“Now I will have less distraction.”

Leonhard EULER [1707 - 1783]