

INFO0902-1 - Structures de données et Algorithmes  
**Projet 3: Algorithmes sur les graphes**

Sébastien Collette (sebastien.collette@ulb.ac.be)

## **Introduction et modalités**

L'objectif de ce projet est d'implémenter des algorithmes simples sur des graphes. Nous vous demandons de remettre le(s) code(s) source correspondant aux desideratas repris ci-dessous, ainsi qu'un bref rapport expliquant ce que vous avez fait et les problèmes rencontrés. Le tout doit être envoyé par e-mail à l'adresse [sebastien.collette@ulb.ac.be](mailto:sebastien.collette@ulb.ac.be) **au plus tard le 26 mai 2011**.

Comme précédemment, les fichiers en C que vous allez remettre devront être compilables sur les ordinateurs du réseau "candi". Finalement, il s'agit d'un travail **personnel** !

## **1 Structure de données**

Dans ce projet, nous travaillerons sur des graphes dirigés et non-dirigés ayant  $n$  sommets. Ces graphes seront pondérés (poids sur les arêtes) ; les poids représenteront la "distance" entre les sommets. L'implémentation de ces graphes se fera grâce à une matrice d'adjacence de taille  $n \times n$ . Les sommets sont simplement dénotés par des nombres de 1 à  $n$ .

Vous avez le droit de créer toutes les structures de données temporaires que vous jugerez nécessaires, l'objectif étant d'obtenir un algorithme dont le pire cas du temps d'exécution (en notation grand- $O$ ) soit le meilleur possible. Bien entendu, vous écrirez à chaque fois la complexité en fonction de  $n$ , le nombre de sommets du graphe.

## **2 Algorithmes sur des graphes dirigés**

### **2.1 Parcours en profondeur (2 points).**

Implémentez un parcours en profondeur du graphe à partir d'un sommet donné  $x$ .

### **2.2 Sommets à distance proche (2 points).**

Etant donné un rayon  $r$  et un sommet  $x$ , listez tous les sommets à distance au plus  $r$  de  $x$ .

### **2.3 Chemin entre deux sommets (3 points).**

Etant donnés deux sommets  $x$  et  $y$ , trouvez un chemin de  $x$  à  $y$  et affichez-le. Vous obtiendrez la totalité des points même si le chemin obtenu n'est pas le plus court.

### 3 Algorithmes sur des graphes non-dirigés

#### 3.1 Graphe acyclique (3 points).

Déterminez si le graphe donné contient un cycle ou pas, et renvoyez un booléen en fonction du résultat.

#### 3.2 Nombre de composantes connexes (4 points).

Une composante connexe est un ensemble maximal de sommets tels qu'il existe un chemin entre toute paire de sommets. Comptez le nombre de composantes connexes du graphe.

#### 3.3 Arbre couvrant de poids minimal (6 points).

Déterminez la liste des arêtes qui composent un arbre couvrant du graphe dont le poids, c'est-à-dire la somme du poids de toutes ses arêtes, est minimal. Affichez le résultat dans le format qui vous convient.

### 4 Questions Bonus - + 1 point à l'examen final

Vous obtiendrez un point bonus en implémentant l'algorithme qui détermine si un graphe est acyclique dans le cas des graphes *dirigés*. Attention, seuls les cycles dont toutes les arêtes sont dans le bon "sens" comptent !

Vous obtiendrez un second point bonus si votre solution pour le "chemin entre deux sommets" retourne le *plus court* chemin avec une complexité de  $O(n^3)$  ou mieux.