EISTI Ecole Internationale des Sciences du Traitement de l'Information	**ING1-TC : ALGO-CONTENEURS : TD : TYPE GRAPHE**

Rédigé par : Hervé de Milleville	Ref : *ING1-ALGO-CONT-TD-GRAPHE*
A l'intention de : Etudiants des ING1-TC	Créé le : 25/03/2012

1. Introduction

- Dans cette feuille d'exercices, on aborde les premiers algorithmes fondamentaux des graphes :
    - Parcours en profondeur
    - Parcours en largeur
    - Accessibilité entre deux sommets
    - Composantes connexes

2. Opérations de parcours dans un graphe

Dans cet exercice, on considère un graphe non orienté.

2.1 Question 1)

Ecrire un algorithme qui permet de marquer tous les sommets d'un graphe à vrai ou à faux en utilisant un conteneur Map

2.2 Question 2)

On donne deux sommets s1 et s2 du graphe. Ecrire l'algorithme qui permet de vérifier qu'il existe une chaîne reliant s1 à s2. On vous demande d'utiliser le principe de parcours en largeur.

2.3 Question 3)

On donne deux sommets s1 et s2 du graphe. Ecrire l'algorithme qui permet de vérifier qu'il existe une chaîne reliant s1 à s2. On vous demande d'utiliser le principe de parcours en profondeur.

3. Composantes connexes dans un graphe

Dans un graphe, deux sommets sont dans la même composante connexe si et seulement si il existe une chaîne entre ces deux sommets.

3.1 Question 1)

Ecrire une procédure qui permet de trouver les composante connexes d'un graphe non orienté. On définira les composantes connexes à l'aide d'un conteneur Map indicé sur les sommets et à valeurs dans l'ensemble des entiers naturels.

3.2 Question 2)

Transformer la procédure précédente en une fonction qui renvoie en plus le nombre de composantes connexes.

4. Composantes fortement connexes dans un graphe orienté

Dans cet exercice, on s'intéresse à des algorithmes permettant de construire les composantes fortement connexes d'un graphe orienté.

4.1 Algorithme de Tarjan

Cet algorithme prend en entrée un graphe orienté et renvoie une partition des sommets du graphe correspondant à ses composantes fortement connexes.

Le principe de l'algorithme est le suivant : on lance un parcours en profondeur depuis un sommet arbitraire. Les sommets explorés sont placés sur une pile *P*. Un marquage spécifique permet de distinguer certains sommets : les racines des composantes fortement connexes, c'est-à-dire les premiers sommets explorés de chaque composante (ces racines dépendent de l'ordre dans lequel on fait le parcours, elles ne sont pas fixées de façon absolue sur le graphe). Lorsqu'on termine l'exploration d'un sommet racine *v*, on retire de la pile tous les sommets jusqu'à *v* inclus. L'ensemble des sommets retirés forme une composante fortement connexe du graphe. S'il reste des sommets non atteints à la fin du parcours, on recommence à partir de l'un d'entre eux.

1. Ecrire cette fonction
2. Donner la complexité de cet algorithme