université angers
FACULTÉ DES SCIENCES
Unité de formation et de recherche

Licence 3 MPCIE
TP Graphes

TP3 Divers Algorithmes :
Flots maximum, composantes connexes et arbres couvrants

Utilisez la structure de graphe implémentée lors des précédents TPs pour effectuer les exercices.

Exercice 1 : Flots maximum

Nous voulons dans cet exercice implémenter un algorithme de flot maximal c'est à dire qui retourne un flot réalisable maximal entre un noeud source et un noeud puit. Nous nous intéressons particulièrement à l'algorithme de Ford-Fulkerson vu en cours. Pour cela, commencez par réfléchir aux changements à effectuer sur votre classe graphe (modélisation absence d'arc, modélisation flot / capacité...). L'algorithme de Ford-Fulkerson n'est pas très précis, c'est une méthode, il existe différentes versions plus détaillées de cet algorithme, comme celle d'Edmons Karp. Implémentez cet algorithme et testez le sur le graphe "exemple4.gr".

Rappel : Le principe de l'algorithme est de chercher s'il existe un chemin améliorant (i.e. un chemin, allant de la source au puit en considérant les arcs comme non-orienté, pour lequel on peut ajouter du flot si l'arc est orienté dans le "bon" sens et retirer du flot sinon) et d'y faire circuler le flot le plus élevé possible (en ajoutant ce flot pour les arcs orientés dans le "bon" sens, en retirant ce flot sinon)
Notez que vous pouvez séparer ces deux parties, d'abord chercher un chemin améliorant, ensuite modifier ce chemin.

Exercice 2 : Composantes connexes

Nous nous intéressons dans cet exercice à la connexité d'un graphe. On considérera un graphe non orienté. Un graphe est dit connexe si pour chaque paire de sommet (u,v) il existe un chemin de u vers v.
Regardez l'algorithme du parcours en profondeur et demandez vous comment il serait possible de l'adapter pour qu'il retourne un booléen permettant de savoir si le graphe est connexe ou non. Implémenter cette modification.
Dans un deuxième temps, retournez le nombre de composantes connexes du graphe, voire les composantes connexes elles-mêmes.
Testez l'algorithme sur le graphe "exemple5.gr".

Exercice 3 : Arbres couvrants

Nous voulons dans cet exercice implémenter un algorithme d'arbre couvrant minimal, c'est à dire que l'on veut obtenir l'arbre qui passe par tous les noeuds du graphe qui est de poids total minimum. Implémentez l'algorithme de Prim et testez le. Cet algorithme est simple et fonctionne comme suit :
1.Choisir un noeud au hasard (ou en passer un en entrée)
2.Ajouter à l'arbre l'arc de poids minimal qui part d un sommet qui est dans l'arbre à un sommet qui n'y est pas.
3.Refaire (2.) jusqu'à ce que tous les sommets soient dans l'arbre.

Notez que le graphe doit être connexe.