Université Paris 7
Maîtrise d'Informatique – Algorithmique
Examen septembre 2003

Duré de l'examen : 2 heures. Documents autorisés : notes de cours/TD. Le barème est indicatif.

Vous pouvez faire appel aux algorithmes vus en cours et/ou TD.

Problème 1 : (4 points)

Montrez comment on peut utiliser l'algorithme de Floyd-Warshall pour décider si un graphe pondéré (avec des poids entiers) contient un cycle de poids négatif. Justifier votre réponse.

Problème 2 : (5 points)

Justifiez qu'une séquence de m opérations Union-Find sur un ensemble à n éléments s'effectue en temps $O(m \log n)$, si l'union se fait par taille (l'arbre le plus petit devient le fils de la racine de l'arbre le plus grand).

Problème 3 : (6 points)

Un graphe orienté $G = (V, E)$ est dit *faible*, si pour toute paire de sommets $u \neq v$ il y au plus un chemin simple de u à v. (Un chemin est simple s'il passe au plus une fois par chaque sommet.)

1. Donnez un exemple de graphe non-acyclique qui est faible.
2. Décrivez un algorithme efficace qui décide si G est faible. Justifiez la validité et la complexité de votre algorithme.

Problème 4 : (5 points)

Soit $G = (V, E)$ un graphe orienté. Un sommet $v \in V$ s'appelle *source*, si tout sommet $w \in V$ $(v \neq w)$ est accessible à partir de v.

Décrivez un algorithme de temps linéaire qui détermine tous les sommets source de G. Justifiez la validité et la complexité de votre algorithme.