UNIVERSITÉ DE MONTRÉAL

DÉPARTEMENT D'INFORMATIQUE ET DE RECHERCHE OPÉRATIONNELLE

EXAMEN FINAL

IFT 6542 FLOTS DANS LES RÉSEAUX

Professeur: BERNARD GENDRON

Date: jeudi, 6 décembre 2007
Heure: 13:30 – 16:30
Salle: 5441

DIRECTIVES:

AUCUNE DOCUMENTATION AUTORISÉE

1 (25 points)

Composantes fortement connexes

a) Soit un graphe orienté $G = (V, A)$. Montrez que G est fortement connexe si et seulement si pour tout $X \subset V$, $X \neq \emptyset$, $(X, \overline{X}) \neq \emptyset$, où $(X, \overline{X})$ est l'ensemble des arcs traversant la coupe définie par X. **Rappel:** $G = (V, A)$ est fortement connexe si et seulement si pour tout couple de sommets distincts i et j, il existe un chemin de i vers j et un chemin de j vers i. *(10 points)*

b) Expliquez le fonctionnement d'un algorithme permettant d'identifier toutes les composantes fortement connexes d'un graphe $G = (V, A)$ (un pseudocode suffit). **Rappel:** Une composante fortement connexe est un sous-graphe fortement connexe dont le graphe sous-jacent est une composante connexe du graphe sous-jacent à G. *(10 points)*

c) Analysez la complexité de votre algorithme en b). *(5 points)*

2 (15 points)

Arbres de poids minimum

a) Expliquez le fonctionnement de la méthode vorace pour résoudre le problème de l'arbre de poids minimum (l'énoncé de la méthode suffit). *(5 points)*

b) Expliquez le fonctionnement de l'algorithme de Boruvka pour résoudre le problème de l'arbre de poids minimum (un pseudocode suffit). **Indice:** L'algorithme de Boruvka maintient une forêt d'arbres tout au long des itérations. *(5 points)*

c) Expliquez pourquoi cet algorithme est un cas particulier de la méthode vorace. *(5 points)*

3 (20 points)

Plus courts chemins

Considérez le problème des plus courts chemins entre toutes les paires de sommets dans un graphe orienté $G = (V, A)$ ayant des arcs de longueur non négative.

a) Expliquez le fonctionnement de l'algorithme de Floyd pour résoudre ce problème (un pseudocode suffit). *(10 points)*

b) Une autre approche pour résoudre ce problème consiste à lancer n exécutions de l'algorithme de Dijkstra. Est-ce que cette approche est préférable à l'algorithme de Floyd? Justifiez. *(10 points)*

4 **(25 points)**

Flot maximum

Soit un graphe orienté $G = (V, A)$ ayant des capacités $u_{ij} > 0$ sur chaque arc $(i, j) \in A$ et dans lequel on cherche un flot maximum entre une source s et un puits t.

a) Supposons qu'un oracle vous fournit le flot maximum entre s et t. Expliquez comment trouver une (s, t)-coupe minimum et analysez la complexité de votre algorithme. **Rappel:** Une (s, t)-coupe $(X, \overline{X})$ est une partition de V en deux sous-ensembles X et $\overline{X}$ telle que $s \in X$ et $t \in X$. La capacité d'une (s, t)-coupe est la somme des capacités des arcs qui traversent la coupe (un arc (i, j) traverse $(X, \overline{X})$ si $i \in X$ et $j \in \overline{X}$). Une (s, t)-coupe est minimum si sa capacité est minimum parmi toutes les (s, t)-coupes de G. *(10 points)*

b) Supposons que vous utilisiez l'algorithme de Ford-Fulkerson pour résoudre le problème du flot maximum entre s et t. Expliquez comment trouver une (s, t)-coupe minimum en utilisant l'information obtenue lors de la dernière itération de l'algorithme. **Rappel:** À chaque itération de l'algorithme de Ford et Fulkerson, on procède à une augmentation de flot le long d'un chemin de s vers t dans le graphe résiduel $G(x)$, sauf si un tel chemin n'a pu être identifié, auquel cas l'algorithme s'arrête. *(5 points)*

c) Soit d un vecteur de *distances* de dimension $n = |V|$ et défini ainsi: $d(t) = 0$ et $d(i)$ est le plus petit nombre d'arcs dans tout chemin de $i \neq t$ vers t dans $G(x)$ (s'il n'y a pas de chemin de $i \neq t$ vers t dans $G(x)$, on pose $d(i) = n$). En supposant que les distances d sont initialisées avec $G(x) = G$, considérez la variante de l'algorithme de Ford et Fulkerson consistant à mettre à jour $G(x)$ et d après chaque augmentation de flot et, lors de l'identification d'un chemin de s vers t dans $G(x)$, à choisir comme prochain sommet non marqué à balayer celui ayant la plus petite distance. En utilisant un tableau *nombre(k)*, $0 \leq k \leq n - 1$, qui donne le nombre de sommets i tels que $d(i) = k$, expliquez comment trouver une (s, t)-coupe minimum en utilisant l'information obtenue lors de la dernière itération de l'algorithme. *(10 points)*

5 **(15 points)**

Transformations entre problèmes

a) Montrez que le problème de flot maximum est un cas particulier du problème de flot à coût minimum. *(5 points)*

b) Montrez que le problème de couplage maximum dans un graphe biparti est un cas particulier du problème de flot maximum. *(5 points)*

c) Montrez que le problème du couplage de poids maximum dans un graphe biparti est un cas particulier du problème d'affectation. *(5 points)*