

UNIVERSITÉ DE MONTRÉAL

DÉPARTEMENT D'INFORMATIQUE ET DE RECHERCHE OPÉRATIONNELLE

EXAMEN FINAL

IFT 6542 FLOTS DANS LES RÉSEAUX

Professeur: BERNARD GENDRON

Date: mercredi, 6 décembre 2006

Heure: 15:30 – 18:30

Salle: 5441

DIRECTIVES:

AUCUNE DOCUMENTATION AUTORISÉE

1 _____ (25 points)

Arbres de poids minimum

- a) Expliquez le fonctionnement de l'algorithme de Prim pour résoudre le problème de l'arbre de poids minimum (un pseudocode suffit). **Indice:** L'algorithme de Prim maintient un seul arbre tout au long des itérations. (5 points)
- b) Considérez une classe de graphes pour lesquels les poids des arêtes sont des valeurs entières dans l'intervalle $[1, k]$, où k est une constante à valeur entière. Proposez une implantation de l'algorithme de Prim permettant de résoudre le problème de l'arbre de poids minimum pour un graphe appartenant à cette classe (précisez les structures de données utilisées). (10 points)
- c) Analysez la complexité de votre implantation en b). (5 points)
- d) En comparaison avec un graphe ayant des poids quelconques sur les arêtes, pouvez-vous résoudre le problème de l'arbre de poids minimum de manière plus efficace pour un graphe appartenant à la classe définie en b)? Justifiez. (5 points)

2 _____ (20 points)

Plus courts chemins

Considérez le problème des plus courts chemins d'un sommet s vers tous les autres dans un graphe orienté $G = (V, A)$ ayant des arcs de longueur négative, mais aucun circuit négatif.

- a) L'algorithme de Dijkstra identifie-t-il correctement les plus courts chemins dans un tel graphe? Si oui, démontrez-le; sinon, donnez un contre-exemple. (5 points)
- b) Supposons qu'on ajoute la valeur $-c_{\min} > 0$ à la longueur de chaque arc du graphe, où $c_{\min} = \min\{c_{ij} \mid (i, j) \in A\}$. L'algorithme de Dijkstra exécuté sur ce graphe avec longueurs modifiées identifie-t-il correctement les plus courts chemins relativement aux longueurs d'origine? Si oui, démontrez-le; sinon, donnez un contre-exemple. (5 points)
- c) Supposons qu'un oracle vous fournit les longueurs des plus courts chemins d'un sommet $r \neq s$ vers tous les autres. Utilisez cette information pour définir des longueurs modifiées de telle sorte que l'algorithme de Dijkstra identifie correctement les plus courts chemins de s vers tous les autres sommets relativement aux longueurs d'origine. Justifiez. (10 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 . 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.

a) Spécifiez par un pseudocode une méthode permettant d'identifier un chemin de s vers t dans $G(x)$.

Indice: Chaque itération de cette méthode consiste à choisir un sommet non marqué et à le balayer (marquer ses successeurs non marqués). (5 points)

b) 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 $d(i) = n$). Expliquez quel algorithme utiliser pour calculer le vecteur de distances d . Quelle est la complexité de cet algorithme? (5 points)

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.

c) Démontrez qu'entre deux augmentations, soit $d(s)$ conserve sa valeur, soit $d(s)$ augmente. (5 points)

d) Démontrez que le nombre d'augmentations pendant lesquelles $d(s)$ conserve sa valeur ne peut dépasser $m = |A|$. (5 points)

e) Déduisez de b), c) et d) la complexité de cette variante de l'algorithme de Ford et Fulkerson. (5 points)

Flot à coût minimum

- a) Formulez le problème du flot à coût minimum dans un graphe orienté $G = (V, A)$ à l'aide d'un modèle de programmation linéaire utilisant les variables x_{ij} qui représentent le flot sur chaque arc $(i, j) \in A$ (le vecteur $x = (x_{ij})_{(i,j) \in A}$ est appelé *flot dans l'espace des arcs*). (5 points)
- b) Soit \mathcal{P} l'ensemble de tous les chemins élémentaires dans G et \mathcal{W} l'ensemble de tous les circuits élémentaires dans G . Soit un *flot dans l'espace des chemins et des circuits* représenté par le vecteur $f = ((f^p)_{p \in \mathcal{P}}, (f^w)_{w \in \mathcal{W}})$. Donnez une formule permettant de calculer le flot x_{ij} sur chaque arc $(i, j) \in A$ à partir d'un flot f dans l'espace des chemins et des circuits. (5 points)
- c) Il est connu que tout flot x dans l'espace des arcs peut être représenté par un flot f dans l'espace des chemins et des circuits qui satisfait la propriété suivante: chaque chemin $p \in \mathcal{P}$ tel que $f^p > 0$ relie un sommet d'offre à un sommet de demande. En utilisant ce résultat, de même que la formule en b), formulez le problème du flot à coût minimum à l'aide d'un modèle de programmation linéaire utilisant la représentation du flot dans l'espace des chemins et des circuits. (5 points)

Couplages

Soit $G = (V, E)$ un graphe non orienté. Un *couplage* de G est un sous-ensemble $W \subseteq E$ tel que deux arêtes quelconques de W ne sont pas adjacentes. Une *couverture* de G est un sous-ensemble $X \subseteq E$ tel que chaque sommet $i \in V$ est l'extrémité d'au moins une arête de X .

- a) Spécifiez un algorithme permettant de construire une *couverture dégrossie* X à partir d'un couplage W . **Rappel:** Une couverture dégrossie ne contient aucune autre couverture de G . (5 points)
 Dans la suite, X désigne une couverture dérivée d'un couplage W par l'algorithme en a).
- b) Montrez que $|X| = |V| - |W|$. (5 points)
- c) Montrez que, si W est un *couplage maximum*, alors X est une *couverture minimum*. **Rappel :** Un couplage maximum est un couplage ayant le plus grand nombre d'arêtes parmi tous les couplages de G . Une couverture minimum est une couverture ayant le plus petit nombre d'arêtes parmi toutes les couvertures de G . (5 points)