Examen d'Algorithmique - Maitrise d'Informatique

27 Janvier 2000

Durée: 3 heures.
Documents: Seuls sont autorisés les notes manuscrites de cours et TD.
NB: Le barême est indicatif.

Exercice 1 (8 points)

Soit $G = (S, A)$ un graphe orienté et w une fonction de pondération $w: A \to \mathbb{Z}$, où $\mathbb{Z}$ est l'ensemble des entiers (positifs ou négatifs). On considère le problème de, étant donné un sommet s,

- décider s'il existe un circuit de poids négatif accessible à partir de s,
- et si un tel circuit n'existe pas, calculer le poids minimum des chemins entre s et chacun des autres sommets de G.

On considère pour ce problème l'algorithme suivant:

L'algorithme utilise une file **F**. On désigne par n_k une variable qui donne la taille de la file **F** au début de l'étape k. Les procédures *ENFILER(x,F)* et *DÉFILER(F)* sont les procédures usuelles d'insertion et de suppression d'un élément x dans la file **F**.

1. *INITIALISER* (s);
2. $\mathbf{F} \leftarrow \{s\}$;
3. $n_0 \leftarrow 1$; $k \leftarrow 0$;
4. **Tant que F** n'est pas vide et $k \leq |S| - 1$ **faire**
5. $\quad$ *ÉTAPE*(k);
6. $\quad$ $k \leftarrow k + 1$;
7. si **F** n'est pas vide **alors retourner**(Circuit négatif).

La procédure *INITIALISER*(s) est définie de la manière suivante:

INITIALISER(s)

$d_0[s] \leftarrow 0$

pour tout $u \in S \backslash \{s\}$ **faire** $d_0[u] \leftarrow \infty$

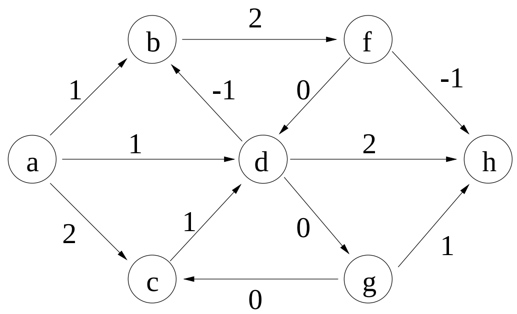

Figure 1: Graphe 1

et la procédure *ÉTAPE*(k) donnée par

ÉTAPE(k)

$n_{k+1} \leftarrow 0$;

$d_{k+1} \leftarrow d_k$;

pour $j = 1$ à n_k **faire**

$x \leftarrow$ *DÉFILER*($\mathbf{F}$);

pour tout successeur y de x **faire**

si $d_{k+1}[y] > d_{k+1}[x] + w(x,y)$ alors

$d_{k+1}[y] \leftarrow d_{k+1}[x] + w(x,y)$;

ENFILER($y, \mathbf{F}$);

$n_{k+1} \leftarrow n_{k+1} + 1$.

Question 1: On dénote par $\mathbf{F_k}$ la file $\mathbf{F}$ au début de l'étape k (avant l'appel *ÉTAPE*(k)). A l'étape 0, on a $\mathbf{F_0} = \{s\}$.

Appliquer l'algorithme ci-dessus aux deux graphes donnés dans les Figures 1 et 2. Donner pour chaque étape $k \geq 0$, les éléments contenus dans la file $\mathbf{F_{k+1}}$, la valeur de n_{k+1}, et les valeurs $d_{k+1}[x]$ qui ont été modifiées par l'étape k.

On dénote par $C_k(x)$ l'ensemble des chemins de s à x de longueur inférieure ou égale à k, et par $\lambda_k(x)$ le poids minimum d'un chemin dans $C_k(x)$. Par convention on pose: $\lambda_k(x) = \infty$ si $C_k(x)$ est vide. On définit $pre(x)$ comme étant l'ensemble des *prédécesseurs* du sommet x dans G, c'est-à-dire:

$$pre(x) = \{y \in S \mid (y,x) \in A\}$$

Question 2: Exprimer $d_{k+1}[x]$ en fonction de $d_k[x]$, pour tout $k \geq 0$, et tout sommet x. (Indication: considérer le fait que la valeur de $d_k[x]$ peut ne pas être modifiée à l'étape k).

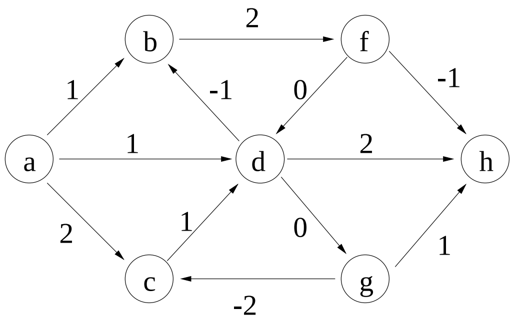

Figure 2: Graphe 2

Question 3: Que peut-on dire si $C_k(x) = \emptyset$? Démontrer que pour tout $k \geq 0$, $d_k[x]$ est égale au poids minimum d'un chemin de $C_k(x)$, c'est-à-dire d'un chemin de longueur inférieure ou égale à k (ici aussi, considérer le fait que la valeur de $d_k[x]$ peut ne pas être modifiée à l'étape k).

Question 4: Démontrer que si $x \in \mathbf{F_k}$, alors $d_k[x]$ est le poids d'un chemin de coût minimum en $C_k(x)$ de longueur exactement k.

Question 5: Prouver que si la file $\mathbf{F}$ n'est pas vide au début de la $|S|$-ième étape ($k = |S|$), alors le graphe contient un circuit de poids négatif accessible à partir de s.

Exercice 2 (8 points)

Soit un graphe orienté $G = (S, A)$. Soit $\sigma = s_0 s_1 s_2 \ldots$ un chemin *infini* de G. On note $Inf(\sigma)$ l'ensemble des sommets qui apparaissent un nombre infini de fois dans σ.

Exemple: Soit $\sigma = s_0 s_1 s_2 s_1 s_2 s_3 s_2 s_3 s_2 s_3 \ldots$ se terminant par une suite infiinie de répétitions de la séquence $s_2 s_3$. Alors, on a $Inf(\sigma) = \{s_2, s_3\}$.

Question 1: Soit C une composante fortement connexe de G. La composante C est *non triviale* si $|C| \geq 2$ ou $C = \{s\}$ avec $(s, s) \in A$. Montrer que, pour toute composante fortement connexe non triviale C, on a:

- pour tout sommet s dans C, il existe un chemin infini σ qui passe par s un nombre infini de fois, c'est-à-dire, $s \in Inf(\sigma)$.
- il existe un chemin infini σ qui passe un nombre infini de fois par *tous* les sommets de C, c'est-à-dire, $C \subseteq Inf(\sigma)$. Est-il vrai dans ce cas que $C = Inf(\sigma)$?

Question 2: Soit $\sigma = s_0 s_1 s_2 \ldots$ un chemin infini de G. Montrer qu'il existe un rang $i \geq 0$ tel que tous les sommets s_j avec $j \geq i$ sont dans une même composante fortement connexe (non triviale), c'est-à-dire, il existe une composante C telle que $\forall j > i$, $s_j \in C$.

Question 3: Soit $F \subseteq S$ un ensemble de sommets, et soit $s \in S$ un sommet particulier dans

S. En supposant donnée une procédure de calcul des composantes fortement connexes d'un graphe, définir un algorithme qui détermine s'il existe un chemin infini σ dans G qui part de s et qui passe un nombre infini de fois par un sommet de F, c'est-à-dire, $Inf(\sigma) \cap F \neq \emptyset$. Donner une preuve de la correction de votre algorithme. Quelle est la complexité de cet algorithme en fonction de la taille du graphe G ?

Question 4: Montrer que $\exists \sigma.\ Inf(\sigma) \cap F \neq \emptyset$ si et seulement si il existe un sommet accessible $s \in F$ qui se trouve dans un circuit. Cela vous suggère-t-il un autre algorithme pour résoudre le problème considéré en Question 3 ? Quelle serait sa complexité ?

Question 5: Soit $V \subseteq S$ un ensemble de sommets. Nous désignons par $pre(V)$ l'ensemble des *prédecesseurs* de V dans G, c'est-à-dire,

$$pre(V) = \{s \in S \mid \exists v \in V.\ (s, v) \in A\}$$

Nous notons par pre^+ la fermeture transitive de pre, et par pre^* la fermeture reflexive-transitive de pre, c'est-à-dire:

$$pre^+(V) = \bigcup_{i>0} pre^i(V)$$

et $pre^*(V) = V \cup pre^+(V)$.

Déduire de la définition ci-dessus des algorithmes itératifs simples pour le calcul de $pre^+(V)$ et $pre^*(V)$, pour un ensemble de sommets V donné. Evaluer leur complexité en fonction de la taille de G.

Question 6: Soit $F \subseteq S$. Comment peut-on calculer:

- l'ensemble des sommets s de S pour lesquels il existe un chemin allant de s à un sommet dans F ?
- l'ensemble des sommets s de F qui se trouvent dans un circuit.

Les réponses doivent être justifiées.

En déduire un algorithme qui calcule l'ensemble des sommets s de S à partir desquels il existe un chemin infini σ passant un nombre infini de fois par un sommet de F, c'est-à-dire, $Inf(\sigma) \cap F \neq \emptyset$. Quelle est la complexité de cet algorithme ?

Exercice 3 (4 points)

Dans cet exercice, on désigne par *fôret* un ensemble d'*arbres dirigés vers la racine*: supposons que les sommets d'une telle fôret sont numérotés de 1 à n. La fôret est définie par une fonction tabulée P qui associe à chaque sommet son *père*. On suppose que pour tout sommet i qui est une *racine*, on a $P(i) = i$.

Question 1: Donner un algorithme parallèle (PRAM) qui, étant donnée une fôret P, calcule une table S telle que, pour tout sommet i, $S(i)$ est la *racine* de l'arbre contenant i. On demande un algorithme dont le temps d'execution est *logarithmique* en fonction de la taille de la fôret. Donner de manière précise la complexité de votre algorithme en *temps* et en *nombre d'opérations*. Dire quel est le modèle PRAM choisi (CRCW, EREW, ou autre).

Question 2: On considère maintenant qu'à chaque sommet i de la fôret est associé un poids

$W(i)$ (W est une table de taille n). Donner un algorithme PRAM qui, étant donnée une fôret P et une fonction de poids W, calcule une table C telle que, pour tout sommet i, $C(i)$ est la somme des poids de tous les sommets se trouvant sur le chemin allant de i vers la racine de son arbre (arbre auquel il appartient).

Comme dans la question précédente, on demande une solution logarithmique. Donner de manière précise la complexité de votre algorithme en *temps* et en *nombre d'opérations*. Dire quel est le modèle PRAM choisi (CRCW, EREW, ou autre).