

Devoir : Magistère Informatique
Algorithmique de graphes
à rendre le jeudi 12 janvier 2012

Michel Habib
habib@liafa.jussieu.fr

13 décembre 2011

Ci-joint quelques questions qui me paraissent intéressantes.

1 Arbres de poids minimum

On considère, un graphe non orienté connexe $G = (X, E)$ et une fonction de poids $\omega : E \rightarrow R_+$

1. **Certificat pour l'arbre de poids minimum** : On considère un graphe $G = (X, E)$ et $T = (X, F)$ une arbre recouvrant de G . Comment vérifier que T est un arbre recouvrant de poids minimum de G ?
Décrire un algorithme avec la meilleure complexité possible. (Piste : il en existe un de complexité linéaire).
2. Peut-on minimiser aussi le diamètre de l'arbre recouvrant de poids minimum? On commencera par montrer qu'il existe des solutions optimales ayant des diamètres différents.
3. * Comment minimiser le nombre de sommets pendants d'un arbre recouvrant de poids minimum?
4. * Comment calculer une base de cycles de poids minimum?

1.1 L'algorithme de Prim vu comme un parcours de graphes

Ecrire une version de l'algorithme de Prim à la manière d'un parcours de graphes avec des Ouverts et des Fermés.

Proposer une caractérisation de l'ordre σ de visite des sommets par un parcours de type Prim.

1.2 Dijkstra

Proposer une caractérisation des ordres de parcours obtenus par l'algorithme de Dijkstra.

2 Parcours en profondeur

Rappelons le programme d'un parcours en profondeur :

```
DFS(G) :  
forall  $v \in X$  do  
     $Ferme(x) \leftarrow Faux$   
end  
forall  $v \in X$  do  
    si  $Ferme(x) = Faux$  alors  
        Explorer(G, x)  
    fin  
end  
Explorer(G, x) ;;  
 $Ferme(x) \leftarrow Vrai$ ;  
pre(x) ;  
forall  $xy \in U$  do  
    si  $Ferme(y) = Faux$  alors  
        Explorer(G, y)  
    fin  
    post(x) ;  
end
```

Et les deux fonctions suivantes utilisant deux variables : comptpre et comptpost étant initialisées à 1.

```
pre(x) :  
 $pre(x) \leftarrow comptpre$  ;  
 $comptpre \leftarrow comptpre + 1$  ;  
  
post(x) :  
 $post(x) \leftarrow comptpost$  ;  
 $comptpost \leftarrow comptpost + 1$  ;
```

1. Montrer que :

Si $post(a) < post(b)$ et $ab \in U$, alors il existe nécessairement un chemin monotone décroissant de b à a dans G , qui n'utilise que des sommets de $[a, b]_{post}$.

2. En déduire que dans ce cas a et b appartiennent à la même composante fortement connexe de G .
3. Est-ce une caractérisation des parcours en profondeur ?
4. Considérons le mot γ associé au parcours en profondeur dans lequel chaque sommet x apparaît exactement deux fois, la première lorsqu'il est exploré $prem(x)$ et la dernière $der(x)$ lorsqu'on dépile x (la fin de l'exploration).

Construire un exemple de graphe ayant plusieurs composantes fortement connexes, y exécuter un parcours en profondeur et calculer l'ordre γ .

5. **Dans ce qui suit on suppose que le parcours en profondeur ne produit pas d'arcs traversiers.**

On va construire sur γ une famille d'intervalles $\mathcal{J}(\mathcal{G})$:

- (a) tous les intervalles $[x, x]$ dont les extrémités sont les deux occurrences du sommet x .
- (b) À chaque arc $ab \in U$ tel que $post(a) < post(b)$, on associe l'intervalle $[der(a), der(b)]$

Deux intervalles I, J se rencontrent si soit

- (a) ils se chevauchent (i.e. $I \cap J \neq \emptyset$ et si $I \not\subseteq J$ et $J \not\subseteq I$).
- (b) l'un est inclus dans l'autre, mais ils ont une extrémité en commun

N.B. L'intersection peut être réduite à un sommet.

Montrer que les sommets extrémités de deux intervalles qui se rencontrent appartiennent à la même composante fortement connexe de G .

6. Il est bien connu que les composantes de chevauchement d'une famille d'ensembles, ordonnées par inclusion constituent une arborescence, appelée l'arborescence de chevauchement. Montrer que les composantes fortement connexes de G correspondent aux composantes de chevauchement de la famille $\mathcal{J}(\mathcal{G})$.
7. Afin de calculer ces composantes il suffit d'exécuter un petit automate qui analyse ces intervalles comme un mot de parenthèses.
8. * Reprendre les questions 5 et 6 si le parcours a engendré des arcs traversiers. Comment caractériser les composantes fortement connexes.

3 Parcours en largeur étendu

On considère un graphe $G = (X, E)$ connexe et à partir d'un sommet initial s on construit les niveaux $L_0 = \{s\}$, $L_1 = N(s)$, ... $L_i = \{\text{sommets à distance exactement } i \text{ de } s\}$, ...

Il est facile d'obtenir ces ensembles de sommets L_i à l'aide d'un simple parcours en largeur de G .

Nous aimerions calculer $\forall i$ les sous-ensembles C_j de L_i vérifiant :

$\forall x, y \in C_j$ il existe une chaîne de x à y n'utilisant que des sommets des niveaux L_k , avec $k \geq i$.

Montrer que ces ensembles permettent de définir un arbre et proposer un algorithme efficace pour les calculer.

4 Chaînes modulo

Cet exercice est plus difficile que les précédents, toutes les idées sont bienvenues. Justifiez votre intuition (polynomial pas polynomial).

1. *** On considère un graphe non orienté $G = (X, E)$, deux sommets $s, t \in X$, quelle est la complexité du problème de décision suivant :
Existe-t-il une chaîne élémentaire de s à t de longueur congrue à i modulo 3 ?
2. * Même question sur un graphe orienté.