Partiel MPRI : Algorithmique de graphes

Michel Habib

26 novembre 2013

NB : Les exercices sont indépendants, * signifie question un peu plus difficile, ** très difficile (enfin pour moi). Tous les graphes G considérés ci-dessous sont non orientés et par commodité n désigne le nombre de sommets et m le nombre d'arêtes).

1 Questions de cours

1. Peut-on implémenter le parcours de graphe MCS (Maximal Cardinality Search) à l'aide de l'affinage de partition ? Si oui, détaillez les opérations et la complexité.
   Rappel de l'algorithme Maximal Cardinality Search :

Data: a graph $G = (V, E)$ and a start vertex s
Result: an ordering σ of V
Assign the label 0 to all vertices
$label(s) \leftarrow 1$
for $i \leftarrow 1$ **à** n **do**
  Pick an unumbered vertex v **with largest label**
  $\sigma(i) \leftarrow v$
  foreach *unnumbered vertex* w *adjacent to* v **do**
    $label(w) \leftarrow label(w) + 1$
  end
end

2. Ecrire un algorithme qui vérifie en temps linéaire qu'un ordre total des sommets donné est simplicial.
3. Ecrire un algorithme qui calcule une clique maximale dominante dans un graphe chordal.
   Rappel : C est une clique dominante de G si $\forall x \in V(G)$, soit $x \in C$ soit x adjacent à C.
4. * Question identique lorsque le graphe n'est pas chordal.
5. Décrire une méthode de calcul pour la treewidth d'un graphe chordal.

2 Parcours LBFS

Rappelons le principe vu en cours du parcours lexicographique en largeur (LBFS).

Lexicographic Breadth First Search (LBFS)

Data: a graph $G = (V, E)$ and a start vertex s
Result: an ordering σ of V
Assign the label $\emptyset$ to all vertices
$label(s) \leftarrow \{n\}$
for $i \leftarrow n$ ***à*** 1 **do**
  Pick an unumbered vertex v **with lexicographically largest label**
  $\sigma(i) \leftarrow v$
  foreach *unnumbered vertex* w *adjacent to* v **do**
    $label(w) \leftarrow label(w).\{i\}$
  end
end

1. Montrer qu'un ordre total τ sur les sommets est un ordre de visite d'un LBFS ssi
   For every triple $a, b, c \in V$ such that $a <_\tau b <_\tau c$ and a is the leftmost of $N(b) \triangle N(c)$ then a $\in N(b) - N(c)$.
   $\triangle$ représente la différence symétrique.
   NB : τ est l'ordre inverse du σ de l'algorithme ci-dessus.
2. Comment certifier qu'un ordre total sur les sommets est bien un ordre de visite d'un parcours LBFS ? Complexité de la vérification de ce certificat.
3. On appelle **ensemble de tie-break**, pour un parcours LBFS, un ensemble de sommets éligibles à une étape de l'algorithme (i.e. ils ont tous une étiquette lexicographiquement maximale). Montrer que si S est un ensemble de tie-break alors la restriction à de τ à S est un ordre de visite légitime d'un LBFS sur $G(S)$.
4. * Montrer que si G est chordal, et S un ensemble de tie-break pour un parcours LBFS, alors pour tout ordre de visite γ d'un autre parcours LBFS sur G, la restriction de γ à S est un ordre de visite légitime d'un LBFS sur $G(S)$.
   On commencera par montrer que cette propriété n'est pas valide sur un graphe quelconque.

3 Maille d'un graphe

On appelle $maille(G)$ la taille du plus petit cycle induit de G. Un cycle induit est un cycle sans corde.

1. Comment calculer la maille d'un graphe chordal ?
2. Proposer un algorithme qui calcule la maille d'un graphe quelconque et évaluer sa complexité.
3. Peut-on appliquer la même technique pour résoudre le problème suivant ?

Data: a graph $G = (V, E)$
Result: G admet un cycle induit de taille k ?

4. Ecrire un algorithme qui répond à la question suivante :
   $maille(G) > 3$? (réponse oui / non).

5. * Que pensez-vous de ce nouveau problème, quelle différence avec celui de la question 3 ?

   Data: un graphe $G = (V, E)$, k un entier
   Result: G admet un cycle induit de taille k ?

4 Graphes sans P_5

Nous avons vu dans le cours plusieurs algorithmes de reconnaissance des cographes (graphes sans P_4), cela peut-il se généraliser ?

Proposer un algorithme de reconnaissance des graphes sans P_5 ayant la meilleure complexité possible.

Cela mérite ** pour un algorithme très efficace.