

ALGORITHMES DISTRIBUÉS

MASTER2 INFORMATIQUE

Novembre 2011

*Ce devoir est à rendre par courriel à gavoille@labri.fr au plus tard **lundi 19 décembre 2011** minuit. Il doit se présenter sous la forme d'un fichier de quelques pages au format .pdf ou .ps comprenant vos nom, prénom et spécialité de Master. Répondez de manière concise tout en justifiant vos réponses. La notation tiendra compte, en premier lieu, de l'argumentation de vos réponses.*

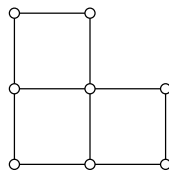
Question 1. Expliquer pourquoi, éventuellement en donnant un exemple, un algorithme distribué déterministe peut avoir plusieurs exécutions différentes, et donc fournir des résultats différents, sur le même graphe et avec les mêmes entrées.

Un ensemble de sommets S d'un graphe G est dit *indépendant* s'il n'y a pas d'arête de G entre deux sommets quelconques de S . Un ensemble indépendant S est *maximal* si dans G il n'existe pas d'ensemble indépendant S' contenant strictement S , c'est-à-dire tel que $S \subset S'$ et $S \neq S'$.

Question 2. Donner un exemple de graphe ayant un ensemble indépendant maximal S et qui possède pourtant un ensemble indépendant S' de taille strictement plus grande que celle de S .

On rappelle qu'une k -coloration pour G est une fonction $c : V(G) \rightarrow \{1, \dots, k\}$ telle que $c(u) \neq c(v)$ si u et v sont voisins. Pour une coloration c de G , on note I_i l'ensemble des sommets de couleur i . Plus formellement $I_i = \{u \in V(G) : c(u) = i\}$.

Question 3. Pour le graphe dessiné ci-dessous, donner une 3-coloration c telle qu'aucun des ensembles I_1, I_2, I_3 ne soit un ensemble indépendant maximal pour G .



Question 4. Démontrer que si G est un graphe connexe possédant une 2-coloration c , alors les ensembles I_1 et I_2 sont tout deux des ensembles indépendants maximaux pour G .

Question 5. Rappeler les caractéristiques du modèle LOCAL.

Tous les algorithmes distribués que l'on considèrera dans la suite seront dans le modèle LOCAL. On se donne un graphe G avec n sommets. On suppose que les sommets de G possèdent une k -coloration, et que pour chaque sommet u , k et la couleur $c(u)$ sont connues de u .

Question 6. Donner un algorithme distribué permettant de calculer un ensemble indépendant maximal pour G . Analyser sa complexité en temps.

L'ensemble S calculé par votre algorithme sera représenté par une variable locale s indiquant pour chaque sommet s'il est dans S ou pas. Plus précisément, pour chaque sommet u , $s(u) \in \{0, 1\}$ et $s(u) = 1$ si et seulement si $u \in S$.

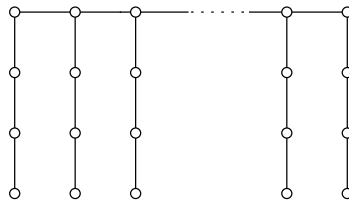
Un ensemble M de sommets de G est r -dominant si pour tout sommet u de G , il existe un sommet $v \in M$ à distance au plus r de u . On rappelle que la distance entre deux sommets d'un graphe est le plus petit nombre d'arête d'un chemin connectant u et v . Un ensemble dominant est simplement un ensemble r -dominant pour $r = 1$. Il n'est pas difficile de voir que tout ensemble indépendant maximal est aussi un ensemble dominant.

Question 7. Dédurre des questions précédantes que tout arbre à n sommets possède un ensemble indépendant maximal de taille au moins $n/2$ et un ensemble dominant de taille au plus $n/2$.

Question 8. Donner un exemple ayant toutes les trois propriétés suivantes :

1. il possède un ensemble dominant M de taille $|M| \geq 3$,
2. il ne possède pas d'ensemble dominant de taille $< |M|$,
3. M n'est pas un ensemble indépendant maximal.

Question 9. Démontrer que tout ensemble 3-dominant M pour l'arbre à n sommets dessiné ci-dessous, un peigne à $b = n/4$ branches avec $b \in \mathbb{N}^*$, est de taille $|M| \geq n/4$.



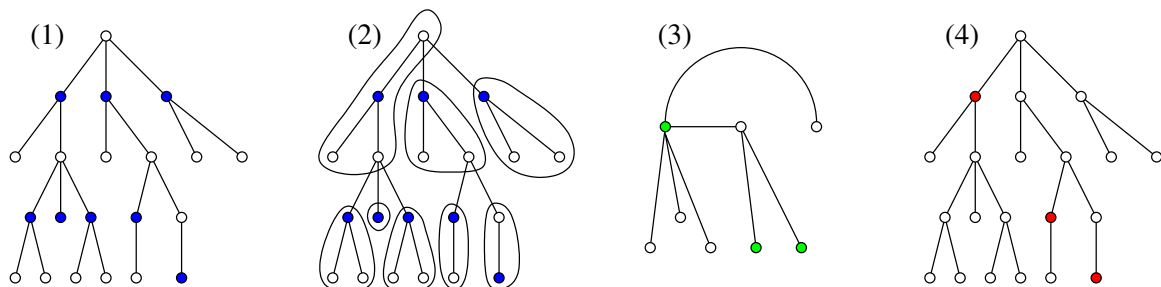
On supposera connu un algorithme distribué SMALLDOM calculant pour tout arbre à n sommets un ensemble dominant M d'au plus $n/2$ sommets et ceci en temps $f(n)$. On supposera que SMALLDOM renvoie en sortie pour chaque sommet u une variable $m(u) \in \{0, 1\}$ telle que $m(u) = 1$ si et seulement si $u \in M$.

On veut concevoir un algorithme distribué SMALLDOM2 permettant de construire un « petit » ensemble 3-dominant ou 4-dominant pour tout arbre à n sommets. L'idée est d'appliquer deux fois la procédure SMALLDOM. Dans la suite un *cluster* est un sous-graphe connexe, ici un sous-arbre.

Plus précisément on réalise les quatres étapes suivantes, chacune illustrée sur la figure ci-dessous :

Algorithme SMALLDOM2 :

- (1) Construire un ensemble dominant M avec SMALLDOM (sommets en bleu).
- (2) Partitionner les sommets en clusters en fonction de M (voir détails ci-après).
- (3) Construire un ensemble dominant M' (en vert) pour l'arbre des clusters avec SMALLDOM.
- (4) Le résultat est $M'' = \{x \in M \cap C \mid C \in M'\}$ (en rouge), les sommets de M appartenant à un cluster de M' .



En plus de former un sous-arbre et une partition des sommets, chaque cluster doit vérifier les deux propriétés suivantes : (P1) Il y a exactement un sommet de M par cluster. (P2) Dans un cluster C , tout sommet $u \notin M$ doit être voisin de l'unique sommet $x \in M \cap C$.

L'arbre des clusters utilisé à l'étape (3) est obtenu en contractant toutes les arêtes internes des clusters formés à l'étape (2).

Question 10. Donner un algorithme distribué permettant de construire de tels clusters à partir d'un ensemble dominant M donné par la variable m . Plus précisément, la sortie de l'algorithme sera, pour chaque sommet u , une variable $c(u)$ indiquant le nom du sommet de M du cluster de u . Dit autrement, si C est un cluster et si x est l'unique sommet de $M \cap C$, alors on doit avoir $c(u) = x$ pour tout $u \in C$.

Question 11. Montrer que SMALLDOM2 calcule un ensemble M'' qui est r -dominant avec $r \leq 4$. Est-ce que « 4 » est la meilleure borne possible pour r ?

Si oui, démontrer que pour tout arbre et toute exécution de SMALLDOM2 on a bien $r \leq 3$. Sinon, donner un contre-exemple M'' où $r \geq 4$, c'est-à-dire l'exemple d'un arbre à n sommets avec un ensemble dominant M de taille $|M| \leq n/2$ où la stratégie adoptée par SMALLDOM2 donnera un sommet à distance 4 de tout sommet de M'' .

Question 12. Montrer que l'ensemble construit par SMALLDOM2 est de taille au plus $n/4$.

Question 13. Donner le pseudo-code pour SMALLDOM2, puis exprimer en fonction de f sa complexité en temps.

FIN.