Master2 Informatique
UE: ALGORITHMIQUE DISTRIBUÉES – 4TIN905U
Responsable : M. Gavoille
Date : 15 janvier 2020
Durée : 1h30
Cours et notes de cours autorisés

Vous devez justifier toutes vos réponses.

On considèrera exclusivement des algorithmes distribués s'exécutant dans le modèle LOCAL. La couleur d'un sommet u sera identifiée par une variable notée $\text{COLOR}(u) \in \mathbb{N}$, tandis que son identifiant par la variable $\text{ID}(u)$.

Sauf mention contraire, les graphes seront considérés comme non orientés. Pour un cycle par exemple, cela signifie qu'il n'y a pas de notion de successeur et de prédécesseur coordonnée entre les sommets. Il n'y a pas de variable $\text{PARENT}(u)$. Dans ces conditions un sommet peut seulement envoyer et recevoir des messages d'un ou plusieurs de ses voisins sans pouvoir déterminer quel est son voisin de droite par exemple.

Coloration des cycles et chemins

Question 1. *Donnez les trois caractéristiques principales du modèle LOCAL. S'il fallait en donner une quatrième, pour le distinguer du modèle à mémoire partagée, quelle serait-elle ?*

On appelle *k-orientation partielle* le fait que chaque sommet u du graphe possède k variables $\text{PARENT}_1(u), \ldots, \text{PARENT}_k(u)$ correspondant chacune à un voisin différent de u ou éventuellement à $\perp$ si la variable en question ne peut être définie, ce qui arrive lorsque u a moins de k voisins. Dans une k-orientation partielle, seuls les quatre cas suivants peuvent se produire, et ce pour chaque arête $u - v$ du graphe :

1. $\exists i, v = \text{PARENT}_i(u)$ et $\exists j, u = \text{PARENT}_j(v)$ (double orientation $u \leftrightarrow v$)
2. $\exists i, v = \text{PARENT}_i(u)$ et $\forall j, u \neq \text{PARENT}_j(v)$ (orientation simple $u \to v$)
3. $\forall i, v \neq \text{PARENT}_i(u)$ et $\exists j, u = \text{PARENT}_j(v)$ (orientation simple $u \leftarrow v$)
4. $\forall i, v \neq \text{PARENT}_i(u)$ et $\forall j, u \neq \text{PARENT}_j(v)$ (arête $u - v$ non orientée)

Notez bien que la différence avec une k-orientation comme vue en cours, est que dans une k-orientation partielle, certaines arêtes se retrouvent non orientées, alors que dans une k-orientation toutes les arêtes sont orientées (et donc le 4e cas ne se produit pas).

Question 2. *Donnez un algorithme distribué pour calculer une k-orientation partielle d'un graphe G en précisant le nombre de ronde.*

Question 3. *Expliquer pourquoi dans un cycle 1-orienté, les arêtes sont nécessairement toutes orientées dans le même sens ?*

On s'intéresse à la 3-coloration des cycles ou chemins non orientés à n sommets. On supposera connu l'algorithme distribué (vu en cours) permettant de 3-colorier un graphe possédant une 1-orientation (toutes les arêtes ont été orientées).

Question 4. *Rappelez le nombre de rondes de cet algorithme, donc celui de 3-coloration d'un graphe possédant une 1-orientation.*

Question 5. *Décrivez un algorithme distribué de 3-coloration pour un cycle ou un chemin non orienté. Précisez le nombre de rondes. [Pensez à l'orienter partiellement et à corriger éventuellement la coloration.]*

Coloration des grilles

On s'intéresse maintenant à la coloration des grilles 2D, avec un total de n sommets. Remarquez que tous les sommets, sauf ceux du bord, sont de degré 4.

Question 6. *Donnez un argument simple expliquant pourquoi toute grille peut être 2-orientée.*

Question 7. *En appliquant la procédure de 2-orientation partielle (question 2) à une grille, que peut-on dire du graphe induit par les arêtes qui restent non orientées ?*

Question 8. *Proposez un algorithme distribué de 9-coloration pour une grille possédant une 2-orientation partielle, puis une version améliorée produisant une 5-coloration. Précisez le nombre de rondes de vos algorithmes.*

Question 9. *Déduire des questions précédentes un algorithme distribué calculant une 5-coloration pour toute grille non orientée. Vous devez précisez la complexité de votre algorithme.*

(a,b)-coloration

On dira qu'un cycle possède une (a, b)-coloration si chaque sommet u possède une couleur $\textsc{Color}(u) \in \{1, 2\}$ de sorte qu'il y ait au plus a sommets de couleur 1 et b sommets de couleur 2 consécutifs sur le cycle. Ainsi une $(1, 1)$-coloration d'un cycle de longueur paire est simplement une 2-coloration classique (alternance des couleurs 1 et 2). Notez que contrairement à une k-coloration classique, une (a, b)-coloration peut avoir deux voisins d'une même couleur dès que $a > 1$ ou $b > 1$.

Question 10. *Montrer qu'une $(1, 2)$-coloration définit un ensemble indépendant maximal, c'est-à-dire un sous-ensemble de sommets deux à deux non adjacents et qui soit maximal pour l'inclusion.*

Question 11. *Donner un algorithme distribué permettant de produire une 3-coloration pour un cycle à partir d'une $(1, 2)$-coloration. Préciser le nombre de rondes qui devra être aussi faible que possible.*

Question 12. *Généraliser votre algorithme de façon à produire une 3-coloration à partir d'une (a, b)-coloration, tout en précisant le nombre de rondes. Vous pourrez supposer que les valeurs a et b sont connues de tous les sommets.*

On admettra la borne inférieure vue en cours sur le nombre de rondes nécessaires pour calculer une 4-coloration dans un cycle à n sommets. Rappelons que la borne inférieure pour un cycle orienté reste valable également pour une 3-coloration d'un cycle non orienté.

Question 13. *À l'aide de cette borne inférieure, montrer que tout algorithme distribué de (a, b)-coloration d'un cycle à n sommets nécessite au moins $(\frac{1}{2} \log^* n) - O(a + b)$ rondes.*

Question 14. *Soient $x, y \in \mathbb{N}$. Montrez que $\max\{x, y\} \geqslant \frac{1}{2}(x+y)$. En déduire que $\log^*(\max\{x, y\}) \geqslant (\log^*(x + y)) - 1$. [Vous pourrez utiliser le fait que $2^{z/2} \geqslant z$ et que $\log^*(2^z) = 1 + \log^*(z)$ pour tout $z \in \mathbb{N}$.]*

Question 15. *À l'aide de cette propriété, montrer que tout algorithme distribué produisant dans un cycle une 3-coloration à partir d'une (a, b)-coloration nécessite au moins $(\frac{1}{2} \log^*(a + b)) - \frac{3}{2}$ rondes.*

FIN.