

SESSION 2009

2nd concours

INFORMATIQUE

École normale supérieure de Lyon

Durée : 3 heures

Ce livret comprend 6 pages numérotées de 1 à 6

L'usage de calculatrices électroniques de poche à alimentation autonome, non imprimantes et sans document d'accompagnement, est autorisé. Cependant, une seule calculatrice à la fois est admise sur la table ou le poste de travail, et aucun échange n'est autorisé entre les candidats.

Graphes de communications

Ce sujet comporte quatre parties. Il est recommandé de lire l'ensemble du sujet avant de commencer la rédaction. Il est également conseillé de traiter les questions dans l'ordre de l'énoncé. On pourra cependant aborder une question en admettant les résultats des questions précédentes. Les algorithmes demandés pourront être écrits en pseudo-code ou dans un langage au choix du candidat, en utilisant les structures de contrôle habituelles.

Notations

Dans la description des algorithmes, nous autorisons les opérations suivantes sur les ensembles :

- affectation : $A \leftarrow B$
- opérations élémentaires : \cup, \cap, \setminus
- itérateur : pour tout x dans A
- test : x appartient à A

Contexte

Un réseau ad hoc est un réseau composé d'objets ou nœuds (ordinateurs, PDAs, téléphones mobiles, *etc.*) pouvant communiquer les uns avec les autres par l'intermédiaire d'une interface radio. Etant données les propriétés d'atténuation du signal radio, un objet ne peut communiquer qu'avec un sous-ensemble des objets du réseau. On dit que deux objets sont *à portée* s'ils peuvent communiquer.

Modélisation

Nous modélisons un réseau ad hoc par un graphe de communications $\mathcal{G} = (V, E)$ dans lequel les sommets $V = \{x_1, \dots, x_{|V|}\}$ correspondent aux objets communicants et les arêtes $E = \{(x_i, x_j), \dots\}$ représentent les communications possibles dans le réseau. En d'autres termes, x_i et x_j peuvent communiquer - sont à portée - si et seulement si $(x_i, x_j) \in E$.

Dans ce sujet, nous considérons uniquement des graphes de communication *symétriques* ou *non-orientés*. C'est à dire que l'on ne différencie pas (x_i, x_j) de (x_j, x_i) . On dit aussi de deux sommets à portée qu'ils sont voisins. Nous notons $N(x_i) = \{x_j | (x_i, x_j) \in E\}$ le voisinage de x_i . Par extension, nous notons $N(A) = \{x_j | \exists x_i \in A, x_j \in N(x_i)\}$ le voisinage d'un ensemble de nœuds A .

Dans les algorithmes, le graphe est donné par la connaissance de l'ensemble des sommets V et des voisinages $N(x), \forall x \in V$.

Partie 1 Chemins et distance

Nous appelons route ou chemin entre x_i et x_j dans \mathcal{G} un ensemble ordonné de sommets $\mathcal{R}(x_i, x_j) = x_{i_1}, x_{i_2}, \dots, x_{i_n}$ tels que $\forall k \in [1..n-1], (x_{i_k}, x_{i_{k+1}}) \in E, x_{i_1} = x_i$ et $x_{i_n} = x_j$. Un graphe est connexe s'il existe un chemin entre chaque paire de sommets. Un sous-ensemble de sommets A est connexe s'il existe un chemin entre chaque paire de sommets de A ne passant que par des sommets de A .

La longueur d'une route \mathcal{R} , notée $l(\mathcal{R})$ est égale au nombre d'arêtes empruntées par \mathcal{R} . Nous définissons la distance entre deux points x_i et x_j , notée $d_{\mathcal{G}}(x_i, x_j)$, comme étant la longueur de la plus petite route reliant x_i à x_j dans \mathcal{G} . Cette distance vaut $+\infty$ s'il n'existe pas de chemin entre x et y . Une route $\mathcal{R}(x_i, x_j)$ est optimale si elle vérifie $l(\mathcal{R}(x_i, x_j)) = d_{\mathcal{G}}(x_i, x_j)$. Une route optimale est aussi appelée un plus court chemin.

Question 1.1. Soit $x, y \in V$ et $z \in N(y)$. Quelle relation y-a-t'il entre $d_{\mathcal{G}}(x, y)$ et $d_{\mathcal{G}}(x, z)$?

Question 1.2. Soit $x, y \in V$. Donner un algorithme permettant de calculer $d_{\mathcal{G}}(x, y)$. Quelle est sa complexité?

Partie 2 Multi-points relais, routage et diffusion

Un ensemble de multi-points relais de x est un sous-ensemble A de $N(x)$ tel que $N(A) \setminus (N(x) \cup \{x\}) = N(N(x)) \setminus (N(x) \cup \{x\})$. On note $MPR(x)$ l'ensemble des ensembles de multi-points relais de x . Un ensemble A de multi-points relais de x est dit optimal si $\forall y \in A, A \setminus \{y\} \notin MPR(x)$. Un ensemble A de multi-points relais de x est dit optimum si $\forall B \in MPR(x), |B| \geq |A|$.

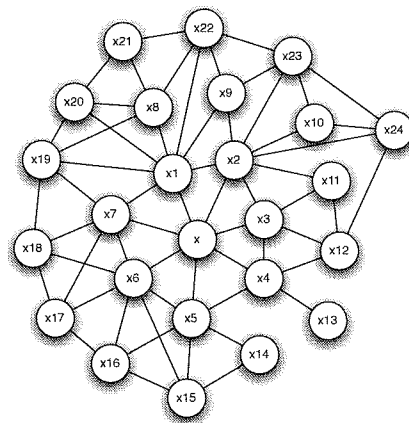


FIGURE 1 – Un exemple de graphe de communications

Question 2.1. Dans la figure 1, donner un ensemble optimum de multi-points relais de x .

Question 2.2. Donner un exemple tel que $|N(x)| > 3$ et $\forall A \in MPR(x), |A| = |N(x)|$

Question 2.3. Donner un exemple tel que $|N(x)| > 3$ et $\exists A \in MPR(x), |A| = 0$.

Question 2.4. Donner un algorithme permettant de calculer un ensemble de multi-points relais optimum. Quelle est sa complexité?

Question 2.5. Donner un algorithme glouton permettant de calculer un ensemble de multi-points relais optimal. Quelle est sa complexité?

Nous supposons maintenant que chaque sommet a choisi ses multi-points relais. Nous associons à un sommet x ses multi-points relais notés $M(x)$. Nous construisons le sous graphe $\mathcal{G}_M(x) = (V, E_M(x))$ avec $E_M(x) = \{(x, y) | y \in N(x)\} \cup \{(x_i, x_j) | x_i \in M(x_j)\}$.

Question 2.6. Montrer que si \mathcal{G} est connexe alors $\mathcal{G}_M(x)$ l'est également.

Question 2.7. Soit $x, y \in V$. Montrer que $d_{\mathcal{G}}(x, y) = d_{\mathcal{G}_M(x)}(x, y)$

Dans la suite du sujet, nous supposons \mathcal{G} connexe. Lorsqu'un nœud x émet un message radio, la nature diffusante du canal radio fait que tous les nœuds de $N(x)$ reçoivent ce message. Par contre, les nœuds appartenant à $V \setminus N(x)$ ne reçoivent pas ce message.

Supposons maintenant que lorsqu'un nœud y reçoit un message, il le ré-émette si les deux conditions suivantes sont respectées :

- c'est la première fois que y reçoit le message ;
- on a $y \in M(z)$ avec z l'émetteur du message que y vient de recevoir.

Question 2.8. Montrer que le message atteint finalement l'ensemble des sommets de \mathcal{G} .

Partie 3 Ensembles dominants

Un sous-ensemble A de V est dit dominant si $\forall x \in V, \exists y \in A, (x, y) \in E$. Nous supposons que chaque nœud possède un identifiant ID distinct compris entre 0 et $|V| - 1$. L'identifiant du nœud v est noté $ID(v)$.

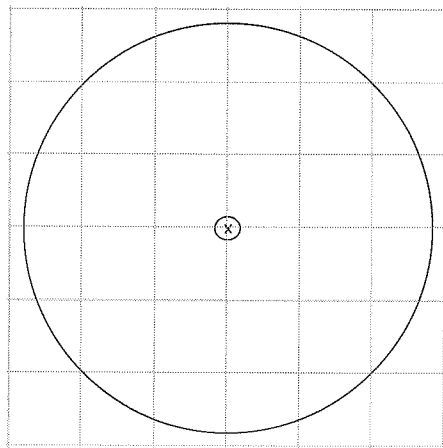


FIGURE 2 – Zone de portée d'un nœud x pavé par une grille de pas $1/2\sqrt{2}$

Nous appliquons un algorithme d'élection à tous les nœuds du réseau. Dans cet algorithme, un nœud x est éliminé si les deux conditions suivantes sont vérifiées :

- l'ensemble Y de ses voisins possédant un identifiant plus petit que lui est connexe ;
- $N(x) \in N(Y)$

On nomme D l'ensemble des nœuds du réseau n'ayant pas été éliminés.

Question 3.1. Montrer que D est un ensemble connexe dominant.

Nous nous plaçons désormais dans le cadre des graphes géométriques aléatoires. Dans ce modèle, les nœuds sont disposés aléatoirement dans un terrain rectangulaire de taille fixée et deux nœuds peuvent communiquer si et seulement s'ils sont à une distance inférieure à 1.

Question 3.2. Montrer que dans le modèle des graphes géométriques aléatoires, la taille moyenne de l'ensemble D reste constante même lorsque le nombre de nœuds augmente.

Indication : étant donné un nœud x , on considérera une grille de pas $1/2\sqrt{2}$ centré sur x (voir la figure 2). On supposera qu'il suffit que chaque carré de côté $1/2\sqrt{2}$ totalement inclus dans la zone de portée de x contienne un élément d'identifiant plus petit que x pour que x s'élimine (en prenant un nœud par carré, on obtient une partie connexe qui couvre la zone de couverture de x).

Partie 4 Coloration

Nous considérons maintenant un sous-ensemble de graphes de communications, les arbres de communications. Un arbre $\mathcal{T} = (V, E)$ est un graphe connexe à $|V| - 1$ arêtes. Parmi l'ensemble des sommets d'un arbre, nous particularisons un sommet que nous appelons *racine*.

Question 4.1. Montrer que pour tout $x \in V, x \neq \text{racine}$, il existe un unique voisin y de x tel que $d_{\mathcal{T}}(\text{racine}, y) < d_{\mathcal{T}}(\text{racine}, x)$. Ce voisin est appelé parent de x et noté $\text{parent}(x)$. Les autres voisins de x sont les fils de x . La racine n'a pas de parent.

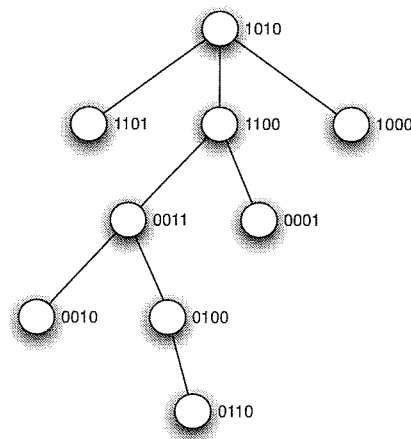


FIGURE 3 – Un arbre de communication et la représentation binaire des identifiants de ses sommets

Colorier un graphe consiste à attribuer à chaque sommet v du graphe une couleur C_v . Un coloriage valide est un coloriage vérifiant la propriété suivante : $\forall (x, y) \in E, C_x \neq C_y$. Un k -coloriage est un coloriage utilisant k couleurs. Dans la suite, les couleurs seront numérotées à partir de 0.

La fonction $\text{bit}_i(C)$ renvoie le i -ème bit de la valeur binaire de la couleur C en commençant par le bit de poids faible dont l'index est 0. Par exemple, $\text{bit}_2(0101) = 1$ et $\text{bit}_0(0101) = 1$. L'opérateur $.$ est l'opérateur de concaténation binaire. Par exemple $101.0 = 1010$.

Question 4.2. Nous supposons que les nœuds de l'arbre illustré figure 3 exécutent l'algorithme 1. Donner la valeur des couleurs des nœuds à la fin de chaque étape de la boucle *tant que*.

Question 4.3. Montrer que l'algorithme 1 construit un coloriage valide.

Algorithme 1 : 6-coloriage d'un arbre

```
début
   $L \leftarrow \lceil \log|V| \rceil$ 
   $C_v \leftarrow ID(v)$ 
  tant que  $L > 3$  et une fois supplémentaire faire
    Envoyer  $(L, C_v)$  à ses fils
    Recevoir  $(L', C_{parent(v)})$ 
    si  $L = L'$  alors
      si  $v$  est la racine alors
         $i_v \leftarrow 0$ 
         $b_v \leftarrow bit_0(C_v)$ 
      sinon
         $i_v \leftarrow \min\{i \mid bit_i(C_v) \neq bit_i(C_{parent(v)})\}$ 
         $b_v \leftarrow bit_{i_v}(C_v)$ 
       $C_v \leftarrow i_v \cdot b_v$ 
       $L \leftarrow \lceil \log L \rceil + 1$ 
fin
```

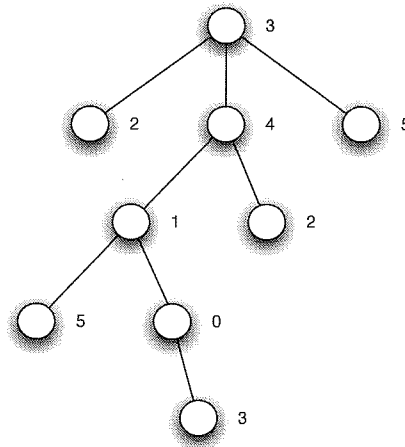


FIGURE 4 – Un arbre de communication et son 6-coloriage

Question 4.4. Montrer que l'algorithme 1 construit un 6-coloriage.

Question 4.5. Nous supposons maintenant que les nœuds de l'arbre illustré figure 4 exécutent l'algorithme 2. Donner la valeur des couleurs des nœuds à la fin de chaque étape de la boucle *pour*.

Question 4.6. Montrer que l'algorithme 2 transforme un 6-coloriage valide en un 3-coloriage valide.

Algorithme 2 : 3-coloriage à partir d'un 6-coloriage

```
début
  pour  $c \leftarrow 5, 4, 3$  faire
    Envoyer  $C_v$  à ses fils
    si  $v$  est la racine alors
      |  $C_v \leftarrow$  plus petite couleur différente de  $C_v$ 
    sinon
      Envoyer  $C_{parent(v)}$  à ses fils
      si  $C_{parent(v)} = c$  alors
        |  $C_v \leftarrow$  plus petite couleur différente de  $C_v$  et  $C_{grandparent(v)}$ 
      sinon
        |  $C_v \leftarrow C_{parent(v)}$ 
fin
```

Fin de l'épreuve.