Complexité
DM
Coloration de graphes

emmanuel.jeandel@lif.univ-mrs.fr
nicolas.ollinger@lif.univ-mrs.fr

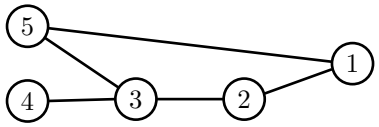
Ce devoir doit être rendu avant le 8 janvier 2010, en version électronique (PDF uniquement) à l'adresse ***emmanuel.jeandel@lif.univ-mrs.fr*** *ou en version papier déposée dans le bureau 319 au CMI, ou encore délivrée en mains propres à E. Jeandel.*

Le devoir doit être effectué par groupe de 1 *à* 3 *personnes, avec au plus un groupe d'une seule personne, chaque personne appartenant à exactement un groupe. Chaque devoir contiendra dans une première partie au moins un paragraphe expliquant comment la répartition du travail s'est effectuée entre les différents membres d'un même groupe. Cette partie n'est facultative que pour les groupes d'une personne et sera prise en compte dans la notation.*

Le sujet de ce DM est l'étude de propriétés liées à la coloration de graphes.

On commence par rappeler certaines définitions.

Dans la suite, un graphe à n sommets est représenté par un tableau T à deux dimensions de sorte que $T[i][j] = 1$ s'il y a une arête de i à j. La taille de l'entrée est donc $N = n^2$.



$$T = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \end{pmatrix} \quad 0100110100010110010010100$$

FIGURE 1 – Un graphe, sa représentation comme un tableau et sa représentation comme un mot

Toutes les complexités dans la suite sont à exprimer en fonction du paramètre N

1 Préliminaires

Q 1) Montrer que le langage suivant est dans **NLOGSPACE**.

$$GAP = \{G | \text{il existe un chemin du premier au deuxième sommet de } G\}$$

Un graphe est dit connexe s'il existe un chemin entre deux paires de sommets quelconques.

Q 2) Montrer que le langage suivant est dans **NLOGSPACE**.

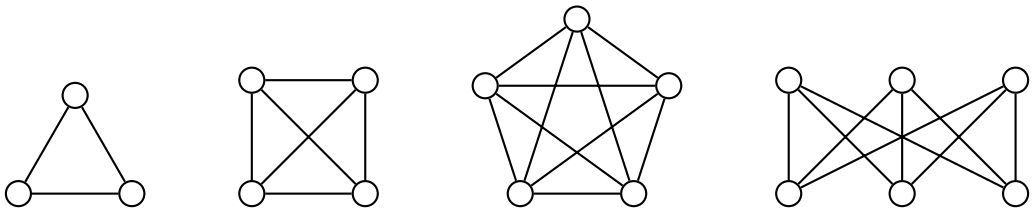
$$CNX = \{G | G \text{ est connexe}\}$$

Le *degré* d'un sommet u d'un graphe G, noté $\Delta(u)$, est le nombre d'arêtes incidentes à u, c-à-d

$$\Delta(u) = \#\{v \in V | T[u][v] = 1\}$$

Le *degré* d'un graphe G, noté $\Delta(G)$, est le degré maximal de ses sommets, c-à-d $\Delta(G) = \max\{\Delta(u) | u \in V\}$.

Un graphe est *k-coloriable* s'il est possible de colorier ses sommets en k couleurs de sorte que deux sommets adjacents aient une couleur différente. Le *nombre chromatique* d'un graphe G, noté $\chi(G)$, est le plus petit nombre de couleurs k tel que G est k-coloriable.



(a) K_3 (b) K_4 (c) K_5 (d) $K_{3,3}$

FIGURE 2 – Graphes à colorier

Q 3) Pour chacun des graphes de la figure 2, calculer le degré du graphe ainsi que son nombre chromatique k, donner un k-coloriage du graphe et une explication de la non existence d'un $(k-1)$-coloriage.

Q 4) Construire pour tout entier k, un graphe G de nombre chromatique $\chi(G) = k$.

Q 5) Montrer que pour tout graphe G, on a $\chi(G) \leqslant \Delta(G) + 1$. Donner un algorithme qui étant donné un graphe G, construit un coloriage avec $(\Delta(G)+1)$ couleurs de G. On analysera sa complexité en temps et en espace en fonction de N.

Pour k un entier quelconque, on note

$$k - COLOR = \{G | G \text{ est } k\text{-coloriable}\}$$

Pour $k = 3$, on obtient donc l'ensemble des graphes 3-coloriables.

Q 6) Montrer que 1-COLOR est dans **LOGSPACE**. Pour cela, on écrira un algorithme dans le langage de son choix fonctionnant en espace logarithmique. Cet algorithme prendra en entrée un tableau T représentant un graphe G. Il répondra 1 si G est dans $1 - COLOR$, et 0 sinon.

Q 7) Montrer que 2-COLOR est dans **P**. Pour cela, on écrira un algorithme dans le langage de son choix fonctionnant en temps polynomial. Cet algorithme prendra en entrée un tableau T représentant un graphe G. Il répondra 1 si G est dans $2 - COLOR$, et 0 sinon.

Q 8) Montrer que 3-COLOR est **NP**-complet. On pourra réutiliser des méthodes vues en TD.

Q 9) Montrer en particulier que si 3-COLOR est dans **P**, alors **P** = **NP**.

Q 10) Montrer que k-COLOR est **NP**-complet pour tout $k > 3$.

2 Morphismes lettre-à-lettre

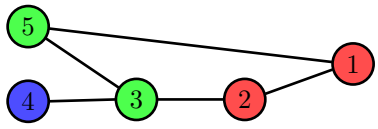
Un morphisme lettre-à-lettre est simplement une fonction h d'un alphabet Σ sur un alphabet Δ qui envoie une lettre de Σ sur une lettre de Δ. Si w est un mot sur l'alphabet Σ, on note $h(w)$ le mot de Δ où on a remplacé toutes les lettres i par leur image $f(i)$.

Par exemple, si $\Sigma = \{a, b, c\}, \Delta = \{a, b\}$ et h la fonction suivante :

$$h : \begin{array}{ccc} a & \mapsto & b \\ b & \mapsto & b \\ c & \mapsto & a \end{array}$$

on a $h(caabacc) = abbbbaa$. Si L est un langage, on note $h(L) = \{h(w), w \in L\}$.

Un graphe pseudo-colorié est un graphe où chaque sommet a une couleur dans l'ensemble $\{R, G, B\}$. On représente toujours le graphe par une matrice T comme précédemment, en mettant en position $T[i][i]$ la couleur du sommet i.



$$T = \begin{pmatrix} R & 1 & 0 & 0 & 1 \\ 1 & R & 1 & 0 & 0 \\ 0 & 1 & G & 1 & 1 \\ 0 & 0 & 1 & B & 0 \\ 1 & 0 & 1 & 0 & G \end{pmatrix} \quad \text{R10011R10001G11001B01010G}$$

FIGURE 3 – Un graphe pseudo-colorié et ses représentations

Un graphe pseudo-colorié est bien colorié si deux sommets reliés par une arête n'ont pas la même couleur.

Q 11) Montrer que le langage suivant est dans **LOGSPACE**. Pour cela, on écrira un algorithme dans le langage de son choix fonctionnant en espace logarithmique. Cet algorithme prendra en entrée un tableau T représentant un graphe G pseudo-colorié. Il répondra 1 si G est dans $BIEN - COL$, et 0 sinon.

$$BIEN - COL = \{G | G \text{ est bien colorié}\}$$

On considère le morphisme suivant :

$$h : \begin{array}{ccc} 0 & \mapsto & 0 \\ 1 & \mapsto & 1 \\ R & \mapsto & 0 \\ G & \mapsto & 0 \\ B & \mapsto & 0 \end{array}$$

Q 12) Montrer que si $G \in BIEN - COL$ alors $h(G) \in 3 - COLOR$

Q 13) Montrer que si $G' \in 3 - COLOR$, alors il existe $G \in BIEN - COL$ tel que $h(G) = G'$.
(*On déduit donc des deux questions précédentes que* $h(BIEN - COL) = 3 - COLOR$)

Q 14) Soit maintenant L un langage quelconque et h un morphisme lettre à lettre. Montrer que si $L \in \mathbf{P}$, alors $h(L) \in \mathbf{NP}$.

Q 15) On suppose que pour tout morphisme h, et pour tout langage $L \in \mathbf{P}$, $h(L) \in \mathbf{P}$. Montrer que $\mathbf{P} = \mathbf{NP}$ (Aide : utiliser la question 9 et la remarque suivant la question 13).

Q 16) Montrer que si $\mathbf{P} = \mathbf{NP}$, alors pour tout morphisme h et pour tout langage $L \in \mathbf{P}$, alors $h(L) \in \mathbf{P}$ (Attention : cette question est facile).

3 Graphes planaires

Dans cette partie, on s'intéresse à la coloration d'une famille particulière de graphes : les graphes planaires.

3.1 Coloration de graphes planaires

Un *plongement* d'un graphe est un dessin de ce graphe sur le plan, c-à-d l'association à chaque sommet d'un point du plan et à chaque arête d'une courbe continue du plan reliant ses sommets incidents. Un plongement est *planaire* si les courbes associées aux arêtes ne se croisent pas. Dans un plongement planaire, les courbes associées aux arêtes partitionnent le plan en un nombre fini de régions appelées *faces*. Un graphe est *planaire* s'il admet un plongement planaire. Pour un graphe planaire donné, on note v le nombre de ses sommets, e le nombre de ses arêtes et f le nombre de ses faces.

Q 17) Pour chacun des plongements représentés sur la figure 2, identifier les plongements planaires. Pour chacun des graphes représentés sur la figure 2, identifier les graphes planaires, dessiner un plongement planaire et compter le nombre de sommets, d'arêtes et de faces.

Q 18) Après avoir remarqué que toute face est entourée par au moins 3 arêtes, montrer que tout graphe planaire connexe satisfait la relation $3f \leqslant 2e$.

Q 19) Par récurrence sur le nombre d'arêtes, montrer que tout graphe planaire connexe satisfait la formule d'Euler : $f - e + v = 2$.

Q 20) Déduire des deux questions précédentes que tout graphe planaire connexe satisfait la relation $e \leqslant 3v - 6$. Qu'en déduit-on pour le graphe K_5 de la figure 2 ?

Q 21) Montrer que tout graphe planaire G possède un sommet u de degré $\Delta(u) < 6$. En déduire que tout graphe planaire est 6-coloriable.

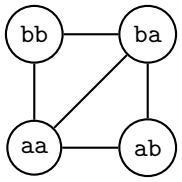
Q 22) Proposer un algorithme qui, étant donné un graphe planaire G, construit un 6-coloriage de G.

3.2 Programmation sur machines de Turing

On souhaite mettre en oeuvre l'algorithme proposé à la question précédente sur machine de Turing. Les machines de Turing considérées ici sont de la variété à plusieurs rubans travaillant sur des rubans infinis à droite ($\delta : Q \times \Gamma^k \to Q \times \Gamma^k \times \{\leftarrow, \downarrow, \rightarrow\}^k$). Pour se faciliter la tâche, il est recommandé d'user de moulte rubans, d'un grand (mais fini) alphabet de travail et de faire bon usage des machines des questions précédentes.

Dans cet exercice, contrairement aux questions précédentes, on représente un graphe sous la forme de listes d'incidence.

Les graphes sont codées sur l'alphabet $\Sigma = \{\texttt{+}, \texttt{-}, \texttt{1}, \texttt{2}, \texttt{3}, \texttt{4}, \texttt{5}, \texttt{6}, \texttt{a}, \texttt{b}, \texttt{(}, \texttt{)}, \texttt{:}, \texttt{,}\}$. À chaque sommet est associé un identifiant unique sur l'alphabet $\{\texttt{a}, \texttt{b}\}$ précédé d'un symbole `+`. Un sommet est codé sous la forme $\texttt{(}u\texttt{:}v_1\texttt{,}\ldots\texttt{,}v_k\texttt{)}$ où u est le code du sommet et v_1, ..., v_k sont les codes des sommets qui lui sont incidents. Un graphe est codé par la suite des codages de ses sommets. Examinons un exemple :



que l'on codera ainsi :

`(+aa:+ab,+ba,+bb)(+ab:+aa,+ba)(+ba:+aa,+ab,+bb)(+bb:+aa,+ba)`

On veut obtenir à la fin un coloriage des sommets du graphe, ce qui revient à transformer les symboles `+` en symboles `1,2,3,4,5,6`. Par exemple, pour le graphe précédent, on pourrait obtenir :

`(1aa:2ab,3ba,4bb)(2ab:1aa,3ba)(3ba:1aa,2ab,4bb)(4bb:1aa,3ba)`

Au cours de l'algorithme, on va utiliser des graphes partiels, partiellement coloriés. On repère par `-` des sommets qu'on a supprimé. Ainsi,

`(+aa:-ab,2ba,+bb)(-ab:+aa,2ba)(2ba:+aa,-ab,+bb)(+bb:+aa,2ba)`

... représente le graphe dans lequel le sommet de code `ab` n'existe plus, et où on a choisi de colorier le sommet de code `ba` avec la couleur 2.

Q 23) Comparer l'efficacité en espace des codages par matrice d'adjacence et par liste d'incidence. Décrire brièvement un algorithme transformant un graphe codé par matrice d'adjacence en un codage de graphe par liste d'incidence. Donner la complexité en temps et en espace de cet algorithme.

Q 24) Expliquer comment vérifier qu'un mot sur Σ est un codage valide de graphe. Donner la complexité en temps et en espace d'un algorithme efficace de vérification.

Q 25) Construire une MT $[q_0|\texttt{COPY}_{i,j}|q_f\rangle$ qui recopie le code de sommet du ruban i sur le ruban j.

Q 26) Expliquer comment coder une pile de sommets. Construire deux MT : $[q_0|\texttt{PUSH}_{i,j}|q_f\rangle$ qui empile le sommet écrit sur le ruban i sur la pile codée sur le ruban j, $[q_0|\texttt{POP}_{i,j}|q_f\rangle$ qui efface le sommet en tête de la pile codée sur le ruban j et l'écrit sur le ruban i

Q 27) Construire une MT $[q_0|\texttt{CMP}_{i,j}|q_y, q_n\rangle$ qui compare les code de sommets des rubans i et j et atteint l'état q_y si les codes sont égaux ou l'état q_n s'ils sont différents.

Q 28) Construire une MT $[q_0|\texttt{SEARCH}_{i,j}|q_y, q_n\rangle$ qui parcourt le code du graphe sur le ruban i à la recherche d'un sommet v de degré inférieur à 6 et atteint l'état q_y si un tel sommet existe ou q_n s'il n'en existe pas. Dans le cas où v existe, son code sera écrit sur le ruban j. (Attention : quand on regarde les sommets reliés au sommet v, on ne tient compte que des sommets non effacés : on ne tient pas compte des sommets indiqués par des symboles `-`)

Q 29) Construire une MT $[q_0|\texttt{ERASE}_{i,j}|q_f\rangle$ qui supprime du graphe codé sur le ruban i le sommet dont le code est indiqué sur le ruban j, ceci en remplaçant les symboles `+` correspondant par des symboles `-`.

Q 30) Construire une MT $[q_0|\texttt{CHOOSE}_{i,j}|q_f\rangle$ qui choisit pour le sommet v indiqué sur le ruban i une couleur différente de celle des sommets reliés à v, et qui écrit sur le ruban j la couleur.

Q 31) Construire une MT $[q_0|\texttt{PAINT}_{i,j,k}|q_f\rangle$ qui parcourt le ruban i et peint toutes les occurrences du sommet indiqué sur le ruban j avec la couleur indiquée sur le ruban k.

Q 32) Combiner les machines précédentes pour construire une MT $[q_0|\texttt{COLOR}_i|q_f\rangle$ qui, étant donné un codage d'un graphe planaire en entrée sur le ruban i, le remplace par un 6-coloriage de ce graphe. Donner la complexité en temps et en espace de cette machine.