Complexité *&* graphes : Contrôle

Université de Tours

Département informatique de Blois

*
* *

Problème 1

Hiérarchiser à l'aide de l'opérateur $\subseteq$ les fonctions de complexité suivantes à l'aide de la notion de O :

1. $\ln(n)$
2. e^n
3. $n^{\log(n)}$
4. n^n
5. $\sqrt{\log(n)}$
6. $n\log(n)$
7. $n!$
8. $2^{n^2\sqrt{\log(n)}}$
9. $\log(\log(n))$
10. $e^{\log(n)^3}$

Justifier les résultats.

Problème 2

L'objectif de cet exercice est de décrire une machine de Turing déterministe avec $\{a, b\}$ comme alphabet d'entrée et # comme symbole blanc.

1. Proposer une machine de Turing qui décale son mot d'entrée initiale d'un caractère vers la droite. Par exemple, si $aab\#$ est le mot en entrée de la machine proposée, après exécution, le mot $\#aab\#$ apparaîtra sur le ruban.

   Vous pourrez notamment introduire un état q_a pour indiquer que le symbole précédemment lu est a et un état q_b pour indiquer que le symbole précédemment lu est b.

   Vous décrirez votre machine sous la forme d'un diagramme d'états en précisant l'état initial et le ou les états accepteurs. b)
2. Déterminer la séquence de configurations de la machine pour le mot d'entrée aab.

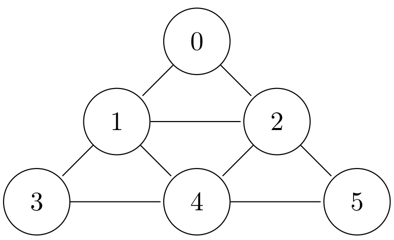
Problème 3

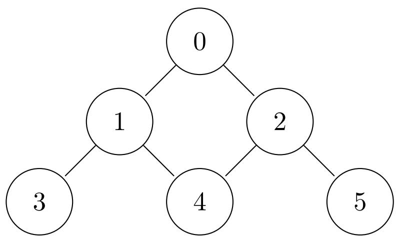
Soit C, un ensemble de couleurs. Colorier un graphe $G = (N, A)$ consiste à appliquer une fonction de coloration $\gamma : N \to C$ qui applique à un noeud donné une couleur.
Une coloration est dire *propre* si toute paire de noeuds voisins a une couleur différente, c'est-à-dire :

$$\forall(x, y) \in A, \gamma(x) \neq \gamma(y)$$

Le nombre minimal de couleurs qu'il faut pour colorier proprement G est appelé *nombre chromatique* et se note $\chi(G)$.
Enfin, étant donné un nombre k, on dit que G est k-coloriable si $\chi(G) \leq k$.

1. Soient les deux graphes G_1 (à gauche) et G_2 (à droite). Montrer que $\chi(G_1) = 3$ et que $\chi(G_2) = 2$.





2. On considère l'algorithme de coloration suivant d'un graphe non orienté.

Créer la liste Γ de taille $|N|$ et telle que $\Gamma[k]$ désigne la couleur associée au noeud n_k. Au départ $\Gamma = [0, 0, ..., 0]$, le graphe n'a aucune couleur.

- Parcourir les noeuds de 1 à $|N|$:
  - Soit n_i, le i-ème noeud à colorier en cours.
  - Colorier n_i avec la plus petite couleur (le plus petit entier) encore non utilisé par ses noeuds adjacents. Si aucun des voisins de n_i n'est colorié, n_i sera colorié par la couleur 1.
- Retourner $\chi(G)$, le nombre chromatique de G.

Proposer une implémentation de cet algorithme (en pseudo code). Vous supposerez disposer d'une matrice d'adjacence M telle que pour tout i et j entre 1 et $|N|$, on a $M[i][j] = 1 \Leftrightarrow (n_i, n_j) \in A$, c'est-à-dire si le noeud n_i est relié au noeud n_j et $M[i][j] = 0$ sinon.

3. Quelle est la complexité en grand O de l'algorithme proposé ?

4. Quel est le nombre chromatique renvoyé par l'algorithme pour G_1 ? Pour G_2 ?

5. La théorie montre que déterminer si un graphe est k-coloriable est NP-complet pour $k > 2$.

   Dans ce contexte, comment interprétez-vous le résultat que vous avez obtenu à la question 4. ? Enfin, quelle réponse peut apporter votre algorithme au problème de décision : un graphe est-il k-coloriable ?

Problème 4

On considère la définition du problème SET-COVER (SC) suivante :

- *Instance* : Un triplet (X, Ω, k) tel que :
  - $X = \{e_1, ..., e_n\}$ est un ensemble d'éléments tel que $\forall i \in [\![1, n]\!], e_i \in E$.
  - $\Omega = \{\Omega_1, ..., \Omega_p\}$ est un ensemble de p sous-ensembles de X tel que $\forall i \in [\![1, p]\!], \Omega_i \subseteq X$.
  - $k \in \mathbb{N}$, un nombre entier.
- *Question* : Existe-t-il un sous-ensemble $I \subseteq \{1, 2, ..., p\}$ d'indices tel que $|I| \leq k$ et $\bigcup_{i \in I} \Omega_i = X$?[1]

Par exemple, soit $X = \{e_1, e_2, e_3, e_4, e_5\}$ et $\Omega = \{\Omega_1, \Omega_2, \Omega_3, \Omega_4\}$ avec $\Omega_1 = \{e_1, e_2\}, \Omega_2 = \{e_1, e_2, e_4\}, \Omega_3 = \{e_4, e_5\}$ et $\Omega_4 = \{e_3\}$.
Pour $k = 3$, la réponse au problème est positive pour $I = \{2, 3, 4\}$, car $\Omega_2 \cup \Omega_3 \cup \Omega_4 = X$.

1. Dans le cadre de l'exemple précédent, avec les mêmes ensembles X et Ω, quelle est la réponse au problème de décision SC pour $k = 2$? Justifiez votre réponse ?

2. Que signifie qu'un problème est NP ? Démontrez que le problème SC est un problème NP.

3. On rappelle la définition du problème VERTEX-COVER (VC) suivante :

   - *Instance* : Un couple (G, k) tel que :
     - $G = (N, A)$, un graphe où N est une ensemble de noeuds $A \subseteq N^2$, un ensemble d'arcs.
     - $k \in \mathbb{N}$, un nombre entier.
   - *Question* : Existe-t-il un sous-ensemble $N' \subseteq N$ tel que $|N'| \leq k$ et $\forall (x, y) \in A, x \in N'$ ou $y \in N'$. Autrement dit, les noeuds de N' couvrent toutes les arrêtes de G.

   (a) En considérant les graphes G_1 (graphe de droite) et G_2 (graphe de gauche) du Problème 2, a t-on $(G_1, 3) \in$ VC ? A t-on $(G_2, 2) \in$ VC ? Si oui, donner les sous-ensembles de noeuds qui permettent de répondre positivement à la question. Sinon, justifiez votre réponse.

[1] On pourra trouver la formulation alternative : Existe-t-il un sous-ensemble $\Omega' \subseteq \Omega$ tel que $|\Omega'| \leq k$ et $\bigcup(\Omega') = X$?.

(b) On propose la fonction de transformation suivante f suivante :

$$f(G,k) = (X^f, \Omega^f, k)$$

où :

- $X^f = A$, l'ensemble d'arcs de G.
- $\Omega^f = \{\Omega^f_1, ..., \Omega^f_{|N|}\}$ avec $\Omega^f_i = \{(x,y) | (x,y) \in A$ et $(x = n_i$ ou $y = n_i)\}$, l'ensemble des arcs dont une extrêmité est le noeud n_i.

En considérant l'instance $(G_2, 2)$, calculer l'instance résultat de $f(G_2, 2)$? A t-on $f(G_2, 2) \in$ SC ?

(c) Démontrer que VC $\leq_p$ SC. On montrera que si $(G,k) \in$ VC, alors $f(G,k) \in$ SC, et réciproquement. Conclure quant à la NP-complétude de SC.