# Année universitaire 2019/20, session 1 d'automne décembre 2019
# Mention : master informatique, code UE 4TIN704U
# Intitulé de l'épreuve : examen Calculabilité et complexité
# Date : 11/12/2019 Heure : 14h30 Durée : 3h
# Document autorisé : 1 feuille A4 recto-verso

**Le barème est indicatif. On attachera une grande importance à la clarté et à la concision des justifications.**

## Question 1 (6 points)

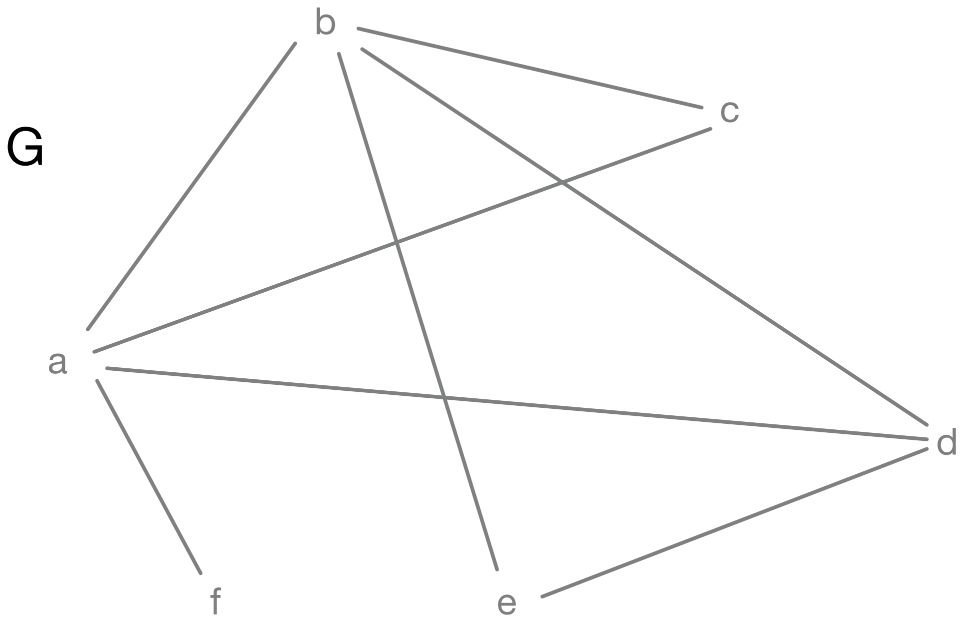
Répondez par "Vrai" ou "Faux" aux questions suivantes. **Détachez et joignez cette première feuille à votre copie, en indiquant en bas votre numéro d'anonymat.** Notation : $0,5$ par réponse juste et $-0,5$ par réponse fausse (l'absence de réponse vaut 0 point). Le résultat final sera le maximum entre 0 et la somme des points réalisés.

*Rappels :* $A \leq B$ signifie que le problème $A$ se réduit au problème $B$ ; $A \leq_P B$ signifie que $A$ se réduit à $B$ par une réduction *polynomiale*.

| | | Vrai | Faux |
|---|---|---|---|
| 1 | $\mathbf{P} \subseteq \mathbf{NP}$ | ☐ | ☐ |
| 2 | Il y a des problèmes dans la classe **NP** qui sont indécidables. | ☐ | ☐ |
| 3 | Si on a deux problèmes $A, B$ tels que $A \leq_P B$, $A \in \mathbf{NP}$ et $B \in \mathbf{P}$, alors $\mathbf{P} = \mathbf{NP}$. | ☐ | ☐ |
| 4 | Si $A \leq_P$ SAT, alors $A$ est un problème dans **NP**. | ☐ | ☐ |
| 5 | On peut trouver des problèmes $A, B$ tels que $A \in \mathbf{NP}$, $B \in \mathbf{P}$ et $A \leq_P B$. | ☐ | ☐ |
| 6 | S'il existe un problème **NP**-difficile qui appartient à **P**, alors $\mathbf{P} = \mathbf{NP}$. | ☐ | ☐ |
| 7 | Si $A$ et son complémentaire $A^{co}$ sont deux problèmes semi-décidables, alors $A$ et $A^{co}$ sont décidables | ☐ | ☐ |
| 8 | On sait décider si un programme LOOP s'arrête sur toutes ses entrées. | ☐ | ☐ |
| 9 | On sait réduire le problème de l'arrêt de programmes WHILE au problème SAT. | ☐ | ☐ |
| 10 | Etant donné un programme WHILE $P$ et deux entiers $m, n$, on sait décider si le calcul de $P$ sur $n$ s'arrête en ayant effectué au plus $m$ instructions. | ☐ | ☐ |
| 11 | Les programmes LOOP et les programmes WHILE calculent les mêmes fonctions. | ☐ | ☐ |
| 12 | Si $A \leq B$ et $A$ est décidable, alors $B$ est décidable. | ☐ | ☐ |

Numéro d'anonymat :

Question 2.1 (a)




U = { e , f }

## Question 2 (5 points)

On considère le problème suivant 3-COL, dont on rappelle qu'il est **NP**-complet :

3-COL

INPUT : Un graphe non-orienté $G$.

QUESTION : Est-ce qu'on peut colorer les sommets de $G$ avec 3 couleurs de telle manière que si deux sommets sont voisins, alors leurs couleurs sont différentes ?

Pour cette question nous allons considérer la variante suivante :

3-COL-POS

INPUT : Un graphe non-orienté $G = (V, E)$, et un sous-ensemble $U \subseteq V$ de sommets.

QUESTION : Est-ce qu'on peut colorer les sommets de $G$ avec 3 couleurs de telle manière que

— si deux sommets sont voisins, alors leurs couleurs sont différentes, et

— tous les sommets de $U$ ont la même couleur.

1. Vous allez montrer que 3-COL-POS se réduit à 3-COL par une réduction polynomiale $f$.
   Soit $G = (V, E), U \subseteq V$ une instance de 3-COL-POS. L'instance $H = f(G, U)$ de 3-COL associée est le graphe $G$ auquel on rajoute 2 nouveaux sommets, appelés $1, 2$ et reliés par une arête, ainsi qu'une arête entre 1 et $u$, et une autre entre 2 et $u$, pour chaque $u \in U$.
   (a) Sur la page 2 vous avez une instance $G, U$ de 3-COL-POS. Complétez sur cette page (que vous allez rendre avec votre copie) le graphe $H = f(G, U)$ associé.
   (b) Justifiez que $G = (V, E), U$ est instance positive de 3-COL-POS si et seulement si $H = f(G, U)$ est instance positive de 3-COL.
   (c) Justifiez que $f$ est une réduction polynomiale.
2. Proposez une réduction polynomiale de 3-COL à 3-COL-POS. (facile)
3. Que déduisez-vous sur la complexité de 3-COL-POS ? Justifiez bien votre réponse.

## Question 3 (5 points)

On considère le problème suivant :

PARTITION

INPUT : Un graphe non-orienté $G = (V, E)$ et un entier $k$.

QUESTION : Peut-on trouver un sous-ensemble $U \subseteq V$ de taille $k$ et tel que chaque sommet dans $U$ a au plus un voisin dans $V \setminus U$ ?

1. Montrez que le problème PARTITION appartient à la classe **NP**, en répondant aux points suivants :
   — quel est le certificat pour une instance positive ? quelle est sa taille ?
   — que fait le vérificateur ? quelle est sa complexité ?

2. <u>On va maintenant construire une réduction polynomiale de PARTITION vers SAT. On ne demande pas que la formule de SAT soit en CNF.</u>

   La formule booléenne $\phi(G, k)$ construite à partir de $G, k$ utilise des variables $x_v$ et $x_{v,i}$, pour tout $v \in V$ et $1 \leq i \leq k$. On veut que $x_{v,i}$ est vraie si et seulement si le sommet $v$ est le $i$-ème sommet de $U$ ; et $x_v$ est vraie si et seulement si $v$ appartient à $U$.

   (a) Expliquez de façon <u>concise</u> pour chacune des formules qui suivent quelle contrainte elle exprime :

      (i) $\bigwedge_{v \in V} \Big( x_v \iff (x_{v,1} \vee x_{v,2} \vee \cdots \vee x_{v,k}) \Big)$

      (ii) $\bigwedge_{1 \leq j \leq k} \bigvee_{v \in V} x_{v,j}$

      (iii) $\bigwedge_{1 \leq j \leq k} \bigwedge_{v,v' \in V, v \neq v'} (\overline{x_{v,j}} \vee \overline{x_{v',j}})$

   (b) Complétez la contrainte qui manque, en la justifiant : $\bigwedge_{u,v,w \in V:\, v \neq w, (u,v) \in E, (u,w) \in E} \ldots$

   (c) La formule $\phi(G, k)$ est la conjonction des formules (a.i), (a.ii), (a.iii), (b). Montrez que $G, k \mapsto \phi(G, k)$ est une réduction polynomiale de PARTITION vers SAT.

   (d) (bonus) Mettez la formule (a.i) en CNF.

## <u>Question 4 (4 points)</u>

Dans cet exercice, « programme » signifie « programme WHILE ». On rappelle que le problème $\mathrm{HALT}_0$ demande si un programme $P$ s'arrête sur l'entrée 0, et que ce problème est indécidable.

Le but de cet exercice est de montrer que le problème suivant est indécidable :

| |
|---|
| <u>BOUCLE</u><br>INPUT : Un programme $R$ avec variables $x_0, \ldots, x_k$.<br>QUESTION : Est-ce qu'il existe des entiers $n_0, \ldots, n_k$ et une instruction $\ell$ tels que le calcul de $R$ à partir de l'instruction $\ell$ avec $x_0 = n_0, \ldots, x_k = n_k$ revient plus tard à l'instruction $\ell$ avec les mêmes valeurs des variables $(x_0 = n_0, \ldots, x_k = n_k)$ ? |

A partir d'un programme $P$ avec variables $x_0, \ldots, x_{k-1}$ on construit les programmes $P_1, P_2$ suivants, avec variables $x_0, \ldots, x_{k-1}, \mathbf{x_k}$ :

— le programme $P_1$ est obtenu de $P$ en rajoutant **après chaque instruction de** $P$ la ligne $\mathbf{x_k} := \mathbf{x_k} + 1$,

— $P_2$ est le programme suivant

$$\textit{WHILE}\ (\text{true})\ \textit{DO}\ (P_1\,;\ x_0 := 0\,; \ldots x_{k-1} := 0\,; \mathbf{x_k} := 0)\ \textit{OD}$$

1. Que vaut la variable $\mathbf{x_k}$ après avoir exécuté $N$ instructions de $P_1$ ?
2. Montrez que le programme $P$ s'arrête sur l'entrée 0 <u>si et seulement si</u> $P_2$ est instance positive du problème BOUCLE. Précisez quelle est l'instruction $\ell$ de $P_2$ et quels sont les entiers $n_0, \ldots, n_k$.
3. Déduisez que le problème BOUCLE est indécidable.