Université d'Aix-Marseille
Master Informatique 1^{ere} année
UE Complexité - Examen partiel - 2022-2023
Durée : 1h30 - Tous documents interdits

Préambule : Pour les questions où vous devez fournir des algorithmes, ceux-ci devront être écrits dans un langage algorithmique avec des instructions de genre "`si alors sinon`", "`tantque`", "`pour`" etc. mais il faudra les définir avec un minimum d'ambiguïté de sorte à ce que le correcteur puisse s'assurer de leur validité. Ainsi, n'utilisez pas les particularités et les spécificités que peuvent avoir certains langages de programmation (par exemple, les boucles "`pour`" sont exprimées de façons différentes en C et en Python ; pour ce type de boucle, il faut préciser exactement les conditions initiales et celles d'arrêt, car sans cela, il est parfois impossible de déterminer la validité d'un algorithme). Quand vous utilisez des boucles, celles-ci doivent impérativement être écrites sur une seule page. Toute structure de données introduite dans vos algorithmes doivent auparavant être explicitées. Il faudra aussi définir avec un maximum de précision vos algorithmes de sorte à ce que l'évaluation de leur complexité soit la plus précise possible (ces évaluations devront toujours être justifiées). Concernant celles-ci, sauf indication contraire, nous utiliserons le modèle de base pour lequel toutes les actions élémentaires, commme les opérations du type tests, affectations ou opérations arithmétiques, sont exécutables en temps constant.

Exercice 1 (2 points). On considère ici des tableaux de n caractères, qui ne peuvent contenir que les caractères "a", "b" ou "B" (blanc), sachant que les caractères "a" ou "b" ne peuvent apparaître que dans le début du tableau, et dès que le caractère "B" apparaît, toutes les cases suivantes contiennent le caractère "B" jusqu'à la fin du tableau. Il vous faut donner ici un algorithme qui prend en entrée un tel tableau et qui affiche "OUI" pour le cas où le nombre de "a" et le nombre de "b" est impair, et qui affiche "NON" dans les autres cas. Donnez une évaluation de la complexité de l'algorithme.

Exercice 2 (4 points). Définissez un programme pour Machine de Turing Déterministe qui reconnaît le langage L défini sur l'alphabet $\{a,b\}$ et constitué des mots qui contiennent un nombre impair de "a" et un nombre impair de "b". On rappelle que pour définir un programme pour Machine de Turing Déterministe, il faut notamment préciser ses états, son alphabet d'entrée ainsi que son alphabet de ruban, et la fonction de transition (vous pouvez fournir celle-ci par un dessin). Donnez une évaluation de la complexité de votre programme. À quelle classe de complexité appartient ce langage L ? Justifiez votre réponse.

Exercice 3 (14 points). On considère ici seulement des graphes non-orientés et sans boucle (donc sans arête reliant un sommet à lui-même). On rappelle qu'une *clique* dans un graphe non-orienté $G = (S,A)$ est un sous-ensemble C de S, i.e. $C \subseteq S$ tel que le sous-graphe de G induit par ces sommets est un graphe complet, i.e. le graphe $G[C]$ est complet, c'est-à-dire que tous les sommets de $G[C]$ sont reliés deux à deux par une arête (notez que le terme *clique* désigne soit le sous-ensemble C, soit le sous-graphe de G induit par C). À partir de cette notion de clique, il est possible de définir le problème de décision suivant :

PARTITION EN CLIQUES

Donnée : Un graphe non-orienté $G = (S,A)$ et un entier $k \leq |S|$.

Question : L'ensemble de sommets S peut-il être partitionné en k sous-ensembles $S_1,S_2,\ldots S_k$ tels que $\forall i, 1 \leq i \leq k$, $G[S_i]$ est un graphe complet, i.e. S_i est une clique ?

On rappelle que si l'ensemble S est partitionné en k sous-ensembles $S_1,S_2,\ldots S_k$, alors on a $\forall i, 1 \leq i \neq j \leq k$, $S_i \cap S_j = \emptyset$ et $\cup_{1 \leq i \leq k} S_i = S$ (avec bien sûr $\forall i, 1 \leq i \leq k$, $S_i \neq \emptyset$).

Question 1 (1 point). Dessinez un graphe non-orienté G de 8 sommets et 12 arêtes tel que le couple $(G,3)$ (i.e. $k = 3$) est une instance positive de *PARTITION EN CLIQUES* et le couple $(G,2)$ (i.e. $k = 2$) est une instance négative de *PARTITION EN CLIQUES*. Dans chaque cas, il vous faut justifier votre réponse.

Question 2 (3 points). À partir du problème de décision *PARTITION EN CLIQUES*, définissez le problème de **recherche** associé, puis celui d'**optimisation** (ici on considère bien sûr comme critère d'optimisation le nombre de cliques), celui de **comptage** et celui d'**énumération**. Enfin, définissez le problème **complémentaire** du problème de décision *PARTITION EN CLIQUES*.

Dans la suite de cet exercice, on supposera que pour un graphe $G = (S,A)$, une partition de $S = \{1,2,\dots n\}$ (les sommets dans S sont représentés par les n premiers entiers non nuls) en k sous-ensembles est représentée par un tableau `t` d'entiers appartenant à l'ensemble $\{1,2,\dots,k\}$, indicé par les sommets de G, tel que $\forall i, 1 \leq i \leq n$, `t[i]=p` si $i \in S_p$ avec donc $p \in \{1,2,\dots,k\}$. Par exemple si $S = \{1,2,3,4,5,6,7,8\}$, une partition de S, en 3 sous-ensembles $S_1 = \{1,3,5\}$, $S_2 = \{2,7\}$ et $S_3 = \{4,6,8\}$ sera représentée par `t[1]=1, t[2]=2, t[3]=1, t[4]=3, t[5]=1, t[6]=3, t[7]=2` et `t[8]=3`.

Question 3 (1 point). Donnez un algorithme qui prend en entrées un graphe non-orienté G de n sommets, un entier k, et un tableau `t` d'entiers indicé de 1 à n, et qui vérifie si `t` représente une partition des n sommets de G en k sous-ensembles. Donnez une évaluation de la complexité de votre algorithme.

Question 4 (4 points). Donnez un algorithme qui prend en entrées un graphe non-orienté G de n sommets, un entier k, et un tableau `t` d'entiers indicé de 1 à n, et qui vérifie si `t` représente une partition des n sommets de G en k sous-ensembles dont chacun est bien une clique de G. Donnez une évaluation de la complexité de votre algorithme. Nous vous demandons d'utiliser à partir de cette question, une représentation des graphes par matrices d'adjacence même si cela engendre une (petite) dégradation de l'efficacité de vos algorithmes.

Question 5 (4 points). Donnez un schéma d'algorithme (il n'est pas nécessaire de donner votre algorithme dans le détail) qui prend en entrées un graphe non-orienté $G = (S,A)$ et un entier k, et teste si S peut être partitionné en $k \leq K$ sous-ensembles $S_1,S_2,\dots S_k$ tels que $\forall i, 1 \leq i \leq k$, $G[S_i]$ est un graphe complet, i.e. S_i est une clique? Donnez une évaluation de la complexité de votre algorithme. Avant de présenter le code de votre algorithme, vous expliquerez clairement la méthode sur laquelle il se base.

Question 6 (1 point). Pouvez-vous affirmer que le problème de décision *PARTITION EN CLIQUES* appartient à la classe de complexité **P**? Justifiez votre réponse.