

Université d'Aix-Marseille - Master Informatique 1^{ère} année

UE Complexité - Exemple de sujet de partiel

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 et 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, comme les opérations du type tests, affectations ou opérations arithmétiques, sont réalisables en temps constant.

Exercice 1. Donnez un algorithme qui prend en entrée un tableau de n caractères, qui ne peut contenir que les caractères "a", "b", ou "B", et qui vérifie si ce tableau contient un mot de la forme $a^k b^k$, où k est un entier non nul, c'est-à-dire si à partir de la première case du tableau, on a une séquence de k cases contenant le caractère "a", suivies de k cases contenant le caractère "b", suivi du caractère "B", s'il reste au moins une case dans le tableau (c'est-à-dire si $2k < n$). Donnez une évaluation de la complexité de l'algorithme.

Exercice 2. Définissez un programme pour Machine de Turing Déterministe qui reconnaît le langage $\{a^k b^k : k > 0\}$ défini sur l'alphabet $\Sigma = \{a, b\}$. Donnez une évaluation de la complexité de votre programme.

Exercice 3. Un *transversal* dans un graphe non-orienté $G = (S, A)$, est un sous-ensemble X de S tel que pour toute arête $\{x, y\} \in A$, alors $x \in X$ ou $y \in X$ (ou non exclusif). En d'autres termes, un transversal est un sous-ensemble de sommets partageant au moins un sommet avec chaque arête du graphe. Vous pourrez utiliser ici soit une représentation de graphes par matrices d'adjacence ou par ou listes d'adjacence.

Question 1. Dessinez un de graphe non-orienté de 6 sommets qui possède un transversal de taille 3 (il a 3 sommets) mais pas de transversal de taille 2.

Question 2. À partir de cette notion de transversal de graphe non-orienté, définissez le problème de décision associé, puis celui de recherche, celui d'optimisation (ici on considère comme critère d'optimisation la taille du transversal) et celui de comptage.

Question 3. Donnez un algorithme qui prend en entrées un graphe non-orienté G et un ensemble de sommets X représenté par un tableau de booléens, et qui vérifie si X est un transversal de G . Donnez une évaluation de la complexité de votre algorithme.

Question 4. Donnez un algorithme qui prend en entrées un graphe non-orienté G et un ensemble de sommets X , et qui vérifie si X est un transversal minimal de G , c'est-à-dire que X ne possède aucun sous-ensemble strict qui soit aussi un transversal. Donnez une évaluation de la complexité de votre algorithme.

Question 5. Donnez un algorithme qui prend en entrée un graphe non-orienté G et qui fournit en résultat un transversal minimal de G . Donnez une évaluation de la complexité de votre algorithme.

Question 6. Donnez un algorithme qui prend en entrées un graphe non-orienté G et un entier k , et teste si le graphe G possède un transversal de taille k ou moins. Donnez une évaluation de la complexité de votre algorithme.