Examen

- 24 Octobre 2023 -
Durée 3 heures.

- Exercice 1 - Pour les langages L suivants sur l'alphabet $\Sigma = \{0, 1\}$, donner un automate fini déterministe acceptant L, ou à défaut montrer qu'il n'en existe pas.

1. L est l'ensemble des mots de longueur impaire.
2. L est l'ensemble des mots de longueur impaire avec 0 comme lettre centrale.
3. On dit que X est un *sous-mot* de Y si on peut obtenir X en effaçant des lettres de Y, par exemple 0110 est sous-mot de 11010101. Le langage L est l'ensemble des mots contenant le sous-mot 0101.
4. L est l'ensemble des mots ne contenant pas le sous-mot 010.

- Exercice 2 - On admet que le problème $SOMME$ qui admet en entrée des entiers $x_1, \ldots, x_n$ et un entier s et retourne VRAI s'il existe un sous-ensemble de $\{x_1, \ldots, x_n\}$ dont la somme est égale à s est NP-complet. On considère le problème $SACADOS$ qui admet en entrée un sac de volume v ainsi que des objets $o_1, \ldots, o_n$, chaque o_i ayant un volume v_i et un prix p_i et qui retourne un sous-ensemble d'objets de prix total maximal dont le volume n'excède pas celui du sac. Montrer que l'existence d'un algorithme polynomial pour $SACADOS$ implique $P = NP$.

- Exercice 3 - On considère un tableau $T[i, j, k]$ de taille $n \times n \times n$ dont les entrées sont $0, 1$. On appelle *couplage3D* un ensemble C de triplets (i, j, k) tels que $T[i, j, k] = 1$ qui diffèrent sur toutes les coordonnées. C'est à dire si $(i, j, k) \in C$ et $(i', j', k') \in C$ sont distincts, alors $i \neq i'$, $j \neq j'$ et $k \neq k'$. Trois sous-ensembles I, J, K de $\{1, \ldots, n\}$ forment un *transversal* si tout triplet (i, j, k) tels que $T[i, j, k] = 1$ vérifie $i \in I$ ou $j \in J$ ou $k \in K$. Sa taille est $|I| + |J| + |K|$.

1. Montrer que la taille d'un transversal minimum est au moins la taille d'un couplage3D.
2. Proposer un algorithme polynomial qui retourne un transversal de taille au plus trois fois la taille d'un transversal minimum.

- Exercice 4 - Soit A un arbre enraciné dont chaque noeud (interne ou feuille) possède un poids entier positif ou nul. Un *indépendant de poids maximum (IPM)* est un ensemble de noeuds deux à deux non reliés de A (aucun n'est un enfant d'un autre) qui maximum le poids total.

1. Proposer un algorithme polynomial basé sur la programmation dynamique calculant un IPM dans un arbre A.

2. * Un graphe est *intercyclique* si tous ses cycles s'intersectent deux à deux. Dans un tel graphe, on sait qu'il existe toujours un ensemble de trois sommets qui intersecte tous les cycles. Proposer un algorithme polynomial calculant un IPM dans un graphe intercyclique dont les sommets sont pondérés.

- Exercice 5 - Décrire un algorithme effectuant le moins de comparaisons possibles calculant dans un tableau:

1. les deux plus petits éléments.
2. le plus petit et le plus grand.

- Exercice 6 - Dans un tableau $T[1,\dots,n]$, une valeur est *majoritaire* si elle apparaît plus de $n/2$ fois. Le problème $MAJORITAIRE$ admet en entrée T et retourne la valeur majoritaire de T ou bien FAUX s'il n'en existe pas.

1. Proposer un algorithme en $O(n \log n)$ pour $MAJORITAIRE$.
2. La probabilité de tirer moins de $n/3$ fois "pile" en effectuant n tirages avec une pièce non biaisée est au plus $1/2^{cn}$ pour une constante $c > 0$. Proposer un algorithme randomisé de résolution de $MAJORITAIRE$ en temps $O(n)$ qui est valide lorsqu'il retourne une valeur majoritaire et se trompe pour les réponses FAUX avec probabilité au plus $1/2^{c\sqrt{n}}$.
3. Utiliser un algorithme vu en cours pour résoudre $MAJORITAIRE$ en temps $O(n)$.
4. * Proposer un algorithme de diviser pour régner en $O(n)$ pour $MAJORITAIRE$. On pourra utiliser un problème auxiliaire $CANDIDATMAJORITAIRE$ retournant x s'il apparaît au plus $k > n/2$ fois dans T et que les autres valeurs apparaissent au plus $n - k$ fois, ou bien retourne FAUX.

- Exercice 7 - Soit $G = (V, E)$ un graphe ayant n sommets et m arêtes supposé codé en espace $\theta(n + m)$.

1. Proposer un algorithme qui teste si G est connexe en temps $\theta(n + m)$.
2. * Proposer un algorithme le plus rapide possible qui teste si G reste connexe même si on efface une de ses arêtes.