Examen terminal ALGO2

9 mai 2022

Écrivez assez grand. Une attention particulière sera accordée à la *rédaction*, au soin et à la rigueur. Les exercices sont indépendants, mais il est conseillé de commencer par les exercices 1 et 2 qui sont proches du cours. Les notes et polycopiés de cours et TD sont autorisés.

1 NP-complétude

On admet que le problème **SUBSET-SUM** est NP-complet.

SUBSET-SUM

ENTRÉES	–	Un ensemble fini $S \subseteq \mathbb{N}$, un objectif $T \in \mathbb{N}$ (donnés en binaire)
QUESTION	–	Existe-t-il un sous-ensemble $X \subseteq S$ de somme exactement T ?

On s'intéresse au problème de décision suivant :

MARATHON IDEAL

ENTRÉES	–	un graphe $G = (V, E)$ non orienté aux arêtes pondérées par $w : E \to \mathbb{N}$, un objectif $L \in \mathbb{N}$ (donnés en binaire)
QUESTION	–	Existe-t-il un cycle C de G utilisant au plus une fois chaque arête de E tel que la somme des poids des arêtes empruntées par C soit exactement L ?

Question 1 Montrer que **MARATHON IDEAL** est NP-complet, en détaillant soigneusement les différentes étapes de la démarche.

Indice : Pour la réduction, écrire $S = \{s_1, \ldots, s_n\}$ et considérer le graphe où, pour aller du sommet 1 au sommet 2, il faut choisir entre une arête de poids 0 et une arête de poids s_1, puis pour aller du sommet 2 au sommet 3, il faut choisir entre une arête de poids 0 et une arête de poids s_2, et ainsi de suite.

2 Programmation linéaire et simplexe

On considère le programme linéaire réel suivant :
$$\text{maximiser } x + 2y \quad \begin{cases} 2x + y \leq 4 \\ 2x - y \geq 0 \\ x, y \geq 0 \end{cases}$$

Question 2 Résoudre ce programme linéaire à l'aide de la méthode du simplexe.

Question 3 Écrire son dual.

3 Algorithme FPT pour d-CENTRE

Soit $n \geq 1$. On travaille sur l'ensemble Σ^n des mots de longueur n sur un alphabet Σ. Pour un mot u de Σ^n, on note $u[i]$ la i-ème lettre de u, de sorte que $u = u[1] \dots u[n]$. Etant donné $u, v \in \Sigma^n$, on définit la distance $\mathsf{d}(u, v)$ comme le nombre de positions auxquelles u et v diffèrent. Formellement :

$$\mathsf{d}(u, v) := |\{i \in \{1, \dots, n\} \mid u[i] \neq v[i]\}|$$

On s'intéresse au problème d-**CENTRE** :

d-**CENTRE**

ENTRÉES	–	k mots $w_1, \dots, w_k$, chacun de longueur n, et un entier d
QUESTION	–	Existe-t-il un mot $u \in A^n$ tel que $\mathsf{d}(u, w_i) \leq d$ pour tout $i \in \{1, \dots, k\}$?

Un tel mot u, s'il en existe un, est appelé un d-centre. Pour une instance $(w_1, \dots, w_k, d)$ de d-**CENTRE**, on dit qu'un indice $h \in \{1, \dots, n\}$ est *mauvais* s'il existe $i, j \in \{1, \dots, k\}$ tels que $w_i[h] \neq w_j[h]$.

Question 4 Que dire s'il existe strictement plus de kd mauvais indices ?

Question 5 En déduire l'existence d'un noyau de taille kd pour d-**CENTRE**. Quelle est la complexité de la construction de ce noyau ?

Question 6 Dans cette question, on suppose qu'il existe un d-centre u, et qu'on a un mot v que l'on sait à une distance au plus $\delta \geq 1$ de u. On suppose que v n'est pas un d-centre : il existe ℓ tel que $\mathsf{d}(v, w_\ell) \geq d + 1$. Il existe donc des indices $i_1 < \dots < i_{d+1}$ auxquels v et w_ℓ diffèrent. Aussi, pour tout $j \in \{1, \dots, d+1\}$, on note v_j le mot obtenu depuis v en remplaçant à l'indice i_j la lettre $v[i_j]$ par $w_l[i_j]$. Montrer qu'au moins un des v_j est à distance au plus $\delta - 1$ de u.

Question 7 Montrer que, s'il existe un d-centre, en partant de w_1 et à l'aide de la procédure de la question précédente, on peut construire un ensemble de $O((d+1)^{d+1})$ mots dans lesquels il y a un d-centre.

Question 8 En déduire un algorithme FPT pour d-**CENTRE** paramétré par d.

4 Sélection paresseuse probabiliste

On considère le problème suivant : **SELECT**

ENTRÉES	–	Un ensemble S de n éléments dans un univers totalement ordonné, un entier k dans $\{1, \dots, n\}$
SORTIE	–	Le k-ème plus petit élément de S noté s_k.

On suppose que tous les éléments de S sont distincts, et que l'on sait comparer deux éléments de S en temps $O(1)$. Par simplicité, on suppose que $n^{3/4}$ et $\sqrt{n}$ sont des entiers. Étant donné un élément a, on note $r_S(a)$ le *rang* de a, c'est-à-dire que a est le $r_S(a)$-ème plus petit élément [1] de S. Le problème **SELECT** consiste donc à donner l'élément de rang k.

On considère l'algorithme probabiliste suivant.

1. $a = s_{r_S(a)}$ où $s_1 < \dots < s_n$ avec $S = \{s_1, \dots, s_n\}$.

```
fonction SelectionParesseuse (S, k)
    R[1..n^{3/4}] := tableau de taille n^{3/4}
    pour i = 1 à n^{3/4} faire
        R[i] := un élément de S choisi uniformément et indépendamment au hasard
    Trier R en place par ordre croissant
    Soit K := kn^{-1/4}
    Soit ℓ := max(K − √n, 1) et m := min(K + √n, n^{3/4})
    Soit a := R[ℓ] et b := R[m]
    calculer r_S(a) et r_S(b)
    si k < n^{3/4} alors
        P := {y ∈ S, y ≤ b}
        decalage := 0
    sinon si k > n − n^{3/4} alors
        P := {y ∈ S, y ≥ a}
        decalage := r_S(a) − 1
    sinon
        P := {y ∈ S, a ≤ y ≤ b}
        decalage := r_S(a) − 1
    si |P| ≥ 4n^{3/4} + 2 ou s_k ∉ P alors échec
    sinon
        Trier P en place par ordre croissant
        renvoyer P[k − decalage]
```

Question 9 Expliquer comment réaliser les instructions soulignés en rouge dans le code, et donner les complexités associées par rapport à n. En déduire la complexité de l'algorithme SelectionParesseuse.

Question 10 Montrer que, lorsque l'algorithme n'échoue pas, il renvoie le bon résultat.

On s'intéresse maintenant à borner la probabilité d'échec de l'algorithme. On note X la variable aléatoire égale au cardinal de $\{i \in \{1, \dots, n^{3/4}\} \mid R[i] \leq s_k\}$.

Question 11 Montrer que $\mathbb{E}(X) = kn^{-1/4}$.

Rappels. La *variance* d'une variable aléatoire Y vaut $V(Y) := \mathbb{E}((Y - \mathbb{E}(Y))^2)$. La variance d'une somme de variables aléatoires indépendantes est la somme des variances. L'*inégalité de Chebychev* dit que, si l'espérance et la variance de Y sont finies, alors pour tout $\alpha > 0$,

$$\mathbb{P}(|Y - E(Y)| \geq \alpha) \leq \frac{V(X)}{\alpha^2}.$$

On suppose désormais que $2n^{3/4} < n$ (vrai pour n suffisamment grand) de façon à ce que les cas $k < n^{3/4}$ et $k > n - n^{3/4}$ soient exclusifs.

Question 12 On suppose que $k \geq n^{3/4}$. Montrer que $\mathbb{P}(a > s_k) = O(n^{-1/4})$.

Un argument symétrique montre que, lorsque $k \leq n - n^{3/4}$, $\mathbb{P}(b < s_k) = O(n^{-1/4})$.

Question 13 En déduire que $\mathbb{P}(s_k \notin P) = O(n^{-1/4})$.

Question 14 Majorer $\mathbb{P}(|P| > 4n^{3/4} + 2)$ par un terme en $O(n^{-1/4})$.

Indice : poser $r_d := max(1, k - 2n^{3/4})$ et $r_u = min(k + 2n^{3/4}, n)$ et montrer qu'il est peu probable que $r_S(a) < r_d$ ou que $r_S(b) > r_u$.

Question 15 En déduire que la probabilité que l'algorithme échoue est en $O(n^{-1/4})$. Que dire du temps d'exécution moyen de l'algorithme qui consiste à recommencer SelectionParesseuse à chaque fois que l'on rencontre un échec ?