Sélection Internationale **Session 2009**
Ecole Normale Supérieure
Epreuve de culture scientifique - **Informatique** Durée : 4 heures

Premier Exercice :

Question-1. On suppose que l'on dispose d'une sous-routine `Merge` qui fusionne deux listes triées de longueurs respectives ℓ et ℓ' en temps $O(\ell+\ell')$. On considère la version ternaire suivante du tri-fusion pour trier une liste $A[0..n-1]$ de n éléments :

`Sort`$(A[0..n-1])$:

1. Si $n \leq 1$, alors renvoyer $A[0..n-1]$
2. Soient $k := \lceil n/3 \rceil$ et $m := \lceil 2n/3 \rceil$.
3. Renvoyer `Merge`(`Sort`$(A[0..k-1])$, `Merge`(`Sort`$(A[k..m-1])$, `Sort`$(A[m..n-1])$)).

Donner (avec preuve) un bon majorant asymptotique du temps de calcul $T(n)$ de `Sort`.

Question-2. Une *pile* est une structure de données opérant comme suit :

1. Vous pouvez tester si la pile est vide ou non ;
2. Vous pouvez toujours rajouter un élément sur le sommet de la pile ;
3. Si la pile n'est pas vide, vous pouvez enlever l'élément au sommet de la pile.

Une *file* est une autre structure de données opérant comme suit :

1. Vous pouvez tester si la file est vide ou non ;
2. Si la file n'est pas vide, vous pouvez enlever l'élément situé au début de la file.
3. Vous pouvez toujours rajouter un élément à la fin de la file.

Montrer comment simuler efficacement une file en utilisant deux piles.

Question-3. Vous recevez un flux contenant exactement $n-1$ nombres distincts de $\{1,\ldots,n\}$: il contient donc tous les nombres de 1 à n, sauf un. Donner un algorithme qui renvoie le nombre manquant, en utilisant au plus $O(\log n)$ bits de mémoire. Vous pouvez supposer que n est connu à l'avance.

Question-4. Vous recevez un flux contenant exactement $n-2$ nombres distincts de $\{1,\ldots,n\}$: il contient donc tous les nombres de 1 à n, sauf deux. Donner un algorithme qui renvoie les deux nombres manquants, en utilisant au plus $O(\log n)$ bits de mémoire. Vous pouvez supposer que n est connu à l'avance.

Second Exercice : Gram-Schmidt à coefficients entiers

On note $\langle , \rangle$ le produit scalaire euclidien de $\mathbb{R}^n$: si $\mathbf{x} = (x_1, \ldots, x_n)$ et $\mathbf{y} = (y_1, \ldots, y_n)$, alors $\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^{n} x_i y_i$. On note $\|.\|$ la norme euclidienne de $\mathbb{R}^n$: si $\mathbf{x} \in \mathbb{R}^n$, alors $\|\mathbf{x}\| = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$.

Soient $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n \in \mathbb{Z}^n$ des vecteurs lignes à coordonnées entières. On suppose que les $\mathbf{b}_i$ sont *linéairement indépendants* : si $\sum_{i=1}^{n} y_i \mathbf{b}_i = 0$ pour un certain $(y_1, \ldots, y_n) \in \mathbb{R}^n$, alors $y_i = 0$ pour tout $i \in \{1, \ldots, n\}$.

On définit de façon itérative les vecteurs lignes $\mathbf{b}_1^*, \mathbf{b}_2^*, \ldots, \mathbf{b}_n^* \in \mathbb{R}^n$ comme suit : $\mathbf{b}_1^* = \mathbf{b}_1$ et pour $2 \leq i \leq n$, $\mathbf{b}_i^* = \mathbf{b}_i - \sum_{j=1}^{i-1} \mu_{i,j} \mathbf{b}_j^*$ où $\mu_{i,j} = \langle \mathbf{b}_i, \mathbf{b}_j^* \rangle / \|\mathbf{b}_j^*\|^2$ pour $1 \leq j < i \leq n$.

Question-1. Montrer que les $\mathbf{b}_i^*$ sont bien définis, c'est-à-dire que la définition de $\mu_{i,j}$ ne comporte pas de division par zéro. Et montrer que les $\mathbf{b}_i^*$ sont orthogonaux deux à deux, c'est-à-dire, si $(i, j) \in \{1, \ldots, n\}^2$ et $i \neq j$, alors $\langle \mathbf{b}_i^*, \mathbf{b}_j^* \rangle = 0$.

Question-2. Soit $d_i = \prod_{j=1}^{i} \|\mathbf{b}_j^*\|^2$ pour $i \in \{1, \ldots, n\}$, et $d_0 = 1$ de sorte que $\|\mathbf{b}_i^*\|^2 = d_i / d_{i-1}$ pour $i \in \{1, \ldots, n\}$. Montrer que pour tout $i \in \{1, \ldots, n\}$, on a $d_i \in \mathbb{Z}$ et $0 < d_i \leq \prod_{j=1}^{i} \|\mathbf{b}_j\|^2$.

Question-3. Montrer que $d_{i-1} \mathbf{b}_i^* \in \mathbb{Z}^n$ pour $1 \leq i \leq n$. Montrer que $\lambda_{i,j} \in \mathbb{Z}$ pour $1 \leq j < i \leq n$ où $\lambda_{i,j} = d_j \mu_{i,j}$.

Question-4. Soit $\lambda_{i,i} = d_i$ pour $0 \leq i \leq n$. Donner un algorithme en temps polynomial (utilisant uniquement de l'arithmétique entière) qui, étant donné en entrée tous les produits scalaires $\langle \mathbf{b}_i, \mathbf{b}_j \rangle$ pour $1 \leq i, j \leq n$, renvoie tous les $\lambda_{i,j}$ pour $1 \leq j \leq i \leq n$. Majorer la complexité de votre algorithme, en supposant $\|\mathbf{b}_i\| \leq B$ pour tout $i \in \{1, \ldots, n\}$, pour un certain $B \in \mathbb{R}$.

Question-5. Donner un algorithme en temps polynomial (utilisant uniquement de l'arithmétique entière) qui, étant donné en entrée $\mathbf{b}_1, \ldots, \mathbf{b}_n$, renvoie $d_{i-1} \mathbf{b}_i^*$ pour $1 \leq i \leq n$. En déduire un algorithme en temps polynomial (utilisant uniquement de l'arithmétique entière) qui, étant donné en entrée $\mathbf{b}_1, \ldots, \mathbf{b}_n$, et un vecteur ligne $\mathbf{b} \in \mathbb{Z}^n$, décide s'il existe $(y_1, \ldots, y_n) \in \mathbb{Z}^n$ tel que $\mathbf{b} = \sum_{i=1}^{n} y_i \mathbf{b}_i$, et dans l'affirmative, renvoie un tel $(y_1, \ldots, y_n)$.

Troisième Exercice : Le problème du logarithme discret
(Exercice conseillé aux candidats de la discipline secondaire)

Soit $(G, \times)$ un groupe cycle fini engendré par $g \in G$. On note $1_G = g^0$ l'élément neutre de G, et n l'ordre de G, ce qui implique que $G = \{1_G, g, g^2, \ldots, g^{n-1}\}$. Soit σ une fonction injective de G vers $\{0,1\}^{\lceil \log_2 n \rceil}$. On suppose que l'on a accès à un oracle $\mathcal{O}$, qui étant donné en entrée $(\sigma(g_1), \sigma(g_2)) \in \sigma(G)^2$, renvoie $\sigma(g_1 g_2^{-1})$.

Question-1. Montrer comment calculer :

1. $\sigma(1_G)$ à partir de n'importe quel $\sigma(h)$ où $h \in G$;
2. un inverse $\sigma(h^{-1})$ à partir de $\sigma(h)$ où $h \in G$;
3. un produit $\sigma(g_1 \times g_2)$ où $(\sigma(g_1), \sigma(g_2)) \in \sigma(G)^2$ est donné en entrée.

Question-2. Donner un algorithme qui, étant donné en entrée $\sigma(h)$ (où $h \in G$) et un entier $e \geq 0$, renvoie $\sigma(h^e)$ en faisant au plus $O(\ell)$ appels à l'oracle $\mathcal{O}$, où ℓ est le nombre de bits de e. Combien d'appels à l'oracle $\mathcal{O}$ votre algorithme fait-il exactement ?

Question-3. Donner un algorithme déterministe qui, étant donné en entrée $(n, \sigma(g), \sigma(g^k))$ où $k \in \{0, 1, \ldots, n-1\}$ est secret, renvoie k en faisant au plus $O(\lceil \sqrt{n} \rceil)$ appels à l'oracle $\mathcal{O}$. Combien d'opérations élémentaires et combien d'espace nécessite votre algorithme ?

Question-4. On suppose donné $(n, \sigma(g), \sigma(g^k))$ où $k \in \{0, 1, \ldots, n-1\}$ est secret et n est premier. Après de nombreux appels à l'oracle $\mathcal{O}$, vous avez obtenu une collection de m triplets $(a_i, b_i, \sigma(g^{a_i + k b_i}))$ où $(a_i, b_i) \in \{0, 1, \ldots, n-1\}^2$, $1 \leq i \leq m$. Montrer que si vous trouvez $(i, j) \in \{1, \ldots, m\}^2$ tel que $i \neq j$, $\sigma(g^{a_i + k b_i}) = \sigma(g^{a_j + k b_j})$ et $(a_i, b_i) \neq (a_j, b_j)$, alors vous pouvez calculer efficacement k.