Sélection Internationale **Session 2008**
Ecole Normale Supérieure
Epreuve de culture scientifique - **Informatique** Durée : 4 heures

Premier Exercice :

Question-1. Soient $\mathcal{R}$ un anneau et $P \in \mathcal{R}[X]$ un polynôme de degré d à coefficients dans $\mathcal{R}$: on note $P(X) = \sum_{i=0}^{d} a_i X^i$ où $a_i \in \mathcal{R}$. Soit $\alpha \in \mathcal{R}$. Donner un algorithme qui calcule $P(\alpha)$ en utilisant au plus $d+1$ multiplications dans $\mathcal{R}$, et $d+1$ additions dans $\mathcal{R}$.

Question-2. Soit $(G, \times)$ un groupe multiplicatif. On suppose que l'on peut multiplier et inverser dans G. Soient $a_1, \ldots, a_n$ des éléments de G. Montrer que l'on peut calculer tous les inverses $a_1^{-1}, a_2^{-1}, \ldots, a_n^{-1}$, en n'utilisant qu'une seule inversion et au plus $3n$ multiplications dans G.

Question-3. Soient a et b des nombres positifs représentés sur ℓ bits. Donner un algorithme qui calcule le plus gros commun diviseur (pgcd) de a et de b en temps quadratique en ℓ, et dont les seules multiplications et divisions utilisées soient des multiplications par deux et des divisions par deux.

Question-4. On admet que l'on peut effectuer en temps polynomial les opérations classiques sur les entiers : addition, soustraction, multiplication et division euclidienne. Montrer alors que l'on peut calculer la partie entière d'une racine carrée d'un entier en temps polynomial : pour tout entier $n \geq 1$ donné en entrée, l'algorithme renvoie $\lfloor \sqrt{n} \rfloor$ en temps polynomial en $\log n$.

Deuxième Exercice : Recherche de cycle

Soit S un ensemble fini. Soit f une fonction de S dans lui-même. Soit $x_0 \in S$. On définit par récurrence $x_i = f(x_{i-1})$ pour tout $i \geq 1$.

Question-1. Montrer qu'il existe un entier $m \geq 0$ tel que $x_i = x_m$ pour une infinité d'indices $i \geq 0$. On note μ le plus petit de ces entiers $m \geq 0$.

Question-2. Montrer qu'il existe un entier $\ell \geq 1$ tel que $x_{\ell+\mu} = x_\mu$. On note λ le plus petit de ces entiers $\ell \geq 1$.

On s'intéresse à l'algorithme suivant pour déterminer le couple (λ, μ) en n'utilisant que très peu d'espace :

Algorithme 1 L'algorithme de Floyd

Entrée : Un élément $x_0 \in S$ et une fonction $f : S \to S$.
Sortie : Le couple (λ, μ) associé.

$a \longleftarrow f(x_0)$
$b \longleftarrow f(a)$
Tant que $a \neq b$ **faire**
    $a \longleftarrow f(a)$
    $b \longleftarrow f(f(b))$
fin Tant que
$m \longleftarrow 0$
$b \longleftarrow a$
$a \longleftarrow x_0$
Tant que $a \neq b$ **faire**
    $a \longleftarrow f(a)$
    $b \longleftarrow f(b)$
    $m \longleftarrow m + 1$
fin Tant que
$\ell \longleftarrow 1$
$b \longleftarrow f(a)$
Tant que $a \neq b$ **faire**
    $b \longleftarrow f(b)$
    $\ell \longleftarrow \ell + 1$
fin Tant que
Renvoyer (ℓ, m).

Question-3. Montrer que l'algorithme 1 termine en un nombre fini d'étapes, et qu'il renvoie bien le couple (λ, μ).

Question-4. Evaluer en fonction de λ et μ le nombre d'appels de la fonction f et le nombre de comparaisons (dans S) effectués par l'algorithme 1.

Question-5. L'algorithme 1 détermine d'abord la valeur de μ, puis celle de λ. Trouver un autre algorithme qui détermine d'abord la valeur de λ, puis celle de μ, sans stocker plus d'éléments de S que l'algorithme 1, ni faire plus d'appels à la fonction f.

Troisième Exercice : Récurrence linéaire dans un corps
(Exercice conseillé aux candidats de la discipline secondaire)

Soit $\mathbb{K}$ un corps. Soit $S = (\alpha_0, \alpha_1, \alpha_2, \dots)$ une suite d'éléments de $\mathbb{K}$. On dit que S est une *suite linéaire* s'il existe $f_0, f_1, \dots, f_{k-1} \in \mathbb{K}$ tels que pour tout entier $i \geq 0$:

$$\alpha_{k+i} = \sum_{j=0}^{k-1} f_j \alpha_{j+i} \tag{1}$$

Un tel k-uplet $(f_0, f_1, \dots, f_{k-1}) \in \mathbb{K}^n$ est alors appelé *rétroaction* de S.

Pour tout polynôme $g \in \mathbb{K}[X]$, on définit l'élément $g \odot S \in \mathbb{K}$ comme :

$$g \odot S = \sum_{j=0}^{d} g_j \alpha_j \tag{2}$$

où d est le degré de g, et les coefficients de g sont donnés par $g = \sum_{j=0}^{d} g_j X^j$ avec $g_j \in \mathbb{K}$. On note $G(S)$ l'ensemble des polynômes $g \in \mathbb{K}[X]$ tels que $(X^i g) \odot S = 0$ pour tout entier $i \geq 0$. Tout élément non nul de $G(S)$ est appelé *polynôme générateur* de S.

Question-1. Montrer que $G(S)$ est un idéal de $\mathbb{K}[X]$.

Question-2. Montrer que S est une suite linéaire si et seulement si $G(S) \neq \{0\}$; et montrer comment calculer une rétroaction de S à partir de n'importe quel polynôme générateur de S.

On suppose désormais que S est une suite linéaire. On appelle *polynôme minimal* de S l'unique générateur de l'idéal $G(S)$ qui soit unitaire, c'est-à-dire de coefficient dominant 1. On note φ ce polynôme minimal, et l'on note m son degré.

Question-3. Soient $g, h \in \mathbb{K}[X]$ deux polynômes. Montrer que si $g - h$ est divisible par φ, alors $g \odot S = h \odot S$.

Question-4. Soit $g \in \mathbb{K}[X]$ un polynôme non nul. Montrer que g est un polynôme générateur de S si et seulement si $(X^i g) \odot S = 0$ pour tout entier $i = 0, \dots, m-1$.

Question-5. En déduire que l'on peut calculer efficacement φ et une rétroaction de S à partir de $\alpha_0, \alpha_1, \dots, \alpha_{2m-1}$. Donner un majorant du nombre d'opérations dans $\mathbb{K}$ utilisées par votre algorithme.

Question-6. Montrer qu'à partir de $2m$ termes consécutifs de la suite S, on peut retrouver les m premiers termes $\alpha_0, \alpha_1, \dots, \alpha_{m-1}$ de la suite S.

Question-7. On définit la série formelle suivante :

$$\alpha = \sum_{i=0}^{\infty} \alpha_i X^{-(i+1)} \in \mathbb{K}((X^{-1})) \tag{3}$$

Soit $g \in \mathbb{K}[X]$. Montrer que $g \in G(S)$ si et seulement si $g\alpha \in \mathbb{K}[X]$.

Question-8. En déduire un nouvel algorithme calculant φ et une rétroaction de S à partir de $\alpha_0, \alpha_1, \ldots, \alpha_{2m-1}$. On pourra utiliser l'algorithme d'Euclide étendu qui, étant donné deux polynômes $g, h \in \mathbb{K}[X]$ et deux entiers r^* et t^* tels que $r^* + t^* \leq \deg(g)$ et $\deg(h) < \deg(g)$, renvoie trois polynômes $r', s', t' \in \mathbb{K}[X]$ tels que $\deg(r') \leq r^*$ et satisfaisant la propriété suivante : pour tous les polynômes $r, s, t \in \mathbb{K}[X]$ tels que $r = sg + th$ avec $\deg(r) < r^*$ et $0 \leq \deg(t) \leq t^*$, il existe un polynôme $u \in \mathbb{K}[X]$ non nul tel que $r = r'u$, $s = s'u$ et $t = t'u$.

Question-9. Entre les deux algorithmes des questions 5 et 8, lequel est le plus efficace ?

Sélection Internationale **Session 2008**
Ecole Normale Supérieure
Scientific Background Assessment - **Computer Science** Duration : 4 hours

First Exercise:

Question-1. Let $\mathcal{R}$ be a ring and $P \in \mathcal{R}[X]$ a polynomial of degree d with coefficients in $\mathcal{R}$, which we denote by $P(X) = \sum_{i=0}^{d} a_i X^i$ where $a_i \in \mathcal{R}$. Let $\alpha \in \mathcal{R}$. Give an algorithm which computes $P(\alpha)$ using at most $d+1$ multiplications in $\mathcal{R}$, and $d+1$ additions in $\mathcal{R}$.

Question-2. Let $(G, \times)$ be a multiplicative group. Assume that one can multiply and invert in G. Let $a_1, \ldots, a_n$ be elements of G. Show that one can compute all the inverses $a_1^{-1}, a_2^{-1}, \ldots, a_n^{-1}$, using only one inversion and at most $3n$ multiplications in G.

Question-3. Let a and b be positive numbers represented with ℓ bits. Give an algorithm which computes the greatest common divisor (gcd) of a and b in time quadratic in ℓ, in such a way that the only multiplications and divisions used are multiplications by two and divisions by two.

Question-4. Assume that one can perform in polynomial time all the classical operations over the integers : addition, subtraction, multiplication and Euclidean division. Show that one can compute the integral part of the square root of an integer in polynomial time : given as input any integer $n \geq 1$, the algorithm outputs $\lfloor \sqrt{n} \rfloor$ in time polynomial in $\log n$.

Second Exercise: Cycle finding

Let S be a finite set. Let f be a mapping from S to S. Let $x_0 \in S$. Define by induction the sequence $x_i = f(x_{i-1})$ for all $i \geq 1$.

Question-1. Show that there exists an integer $m \geq 0$ such that $x_i = x_m$ for infinitely many indices $i \geq 0$. Denote by μ the smallest such integer $m \geq 0$.

Question-2. Show that there exists an integer $\ell \geq 1$ such that $x_{\ell+\mu} = x_\mu$. Denote by λ the smallest such integer $\ell \geq 1$.

We study the following algorithm to compute the couple (λ, μ) using negligible space :

Algorithme 1 Floyd's Algorithm

Input: An element $x_0 \in S$ and a function $f : S \to S$.

Output: The corresponding couple (λ, μ).

```
a ⟵ f(x_0)
b ⟵ f(a)
while a ≠ b do
    a ⟵ f(a)
    b ⟵ f(f(b))
end while
m ⟵ 0
b ⟵ a
a ⟵ x_0
while a ≠ b do
    a ⟵ f(a)
    b ⟵ f(b)
    m ⟵ m + 1
end while
ℓ ⟵ 1
b ⟵ f(a)
while a ≠ b do
    b ⟵ f(b)
    ℓ ⟵ ℓ + 1
end while
Output (ℓ, m).
```

Question-3. Show that Algorithm 1 terminates in finitely many steps, and that it outputs the couple (λ, μ).

Question-4. Evaluate as a function of λ and μ the number of calls of the function f, and the number of comparisons (in S) used by Algorithm 1.

Question-5. Algorithm 1 first computes the value of μ, then that of λ. Find another algorithm which first computes the value of λ, then that of μ, without storing more elements of S than Algorithm 1, nor making more calls to the function f.

Third Exercise: Linearly generated sequence in a field

(Exercise recommended to the candidates of the subsidiary subject)

Let $\mathbb{K}$ be a field. Let $S = (\alpha_0, \alpha_1, \alpha_2, \dots)$ be a sequence of elements in $\mathbb{K}$. The sequence S is said to be a *linear sequence* if there exist $f_0, f_1, \dots, f_{k-1} \in \mathbb{K}$ such that for any integer $i \geq 0$:

$$\alpha_{k+i} = \sum_{j=0}^{k-1} f_j \alpha_{j+i} \tag{1}$$

Such a k-uplet $(f_0, f_1, \dots, f_{k-1}) \in \mathbb{K}^n$ is then called a *feedback* of S.

For any polynomial $g \in \mathbb{K}[X]$, we define the element $g \odot S \in \mathbb{K}$ as :

$$g \odot S = \sum_{j=0}^{d} g_j \alpha_j \tag{2}$$

where d is the degree of g, and the coefficients of g are given by $g = \sum_{j=0}^{d} g_j X^j$ where $g_j \in \mathbb{K}$. Denote by $G(S)$ the set of polynomials $g \in \mathbb{K}[X]$ such that $(X^i g) \odot S = 0$ for all integers $i \geq 0$. Every non-zero element of $G(S)$ is called a *generating polynomial* of S.

Question-1. Show that $G(S)$ is an ideal of $\mathbb{K}[X]$.

Question-2. Show that S is a linear sequence if and only if $G(S) \neq \{0\}$; and show how to compute a feedback of S from any generating polynomial of S.

From now on, we assume that S is a linear sequence. We call *minimal polynomial* of S the unique generator of the ideal $G(S)$ which is monic, that is, whose leading coefficient is equal to 1. We denote by φ this minimal polynomial, and we denote by m its degree.

Question-3. Let $g, h \in \mathbb{K}[X]$ be two polynomials. Show that if $g - h$ is divisible by φ, then $g \odot S = h \odot S$.

Question-4. Let $g \in \mathbb{K}[X]$ be a non-zero polynomial. Show that g is a generating polynomial of S if and only if $(X^i g) \odot S = 0$ for all integers $i = 0, \dots, m-1$.

Question-5. Deduce that one can efficiently compute φ and a feedback of S from $\alpha_0, \alpha_1, \dots, \alpha_{2m-1}$. Give an upper bound on the number of operations in $\mathbb{K}$ used by your algorithm.

Question-6. Show that from any $2m$ consecutive elements of the sequence S, one can recover the first m elements $\alpha_0, \alpha_1, \dots, \alpha_{m-1}$ of the sequence S.

Question-7. We define the following formal series :

$$\alpha = \sum_{i=0}^{\infty} \alpha_i X^{-(i+1)} \in \mathbb{K}((X^{-1})) \tag{3}$$

Let $g \in \mathbb{K}[X]$. Show that $g \in G(S)$ if and only if $g\alpha \in \mathbb{K}[X]$.

Question-8. Deduce a new algorithm which computes φ and a feedback of S from $\alpha_0, \alpha_1, \ldots, \alpha_{2m-1}$. One might use the extended Euclidean algorithm which, given as input two polynomials $g, h \in \mathbb{K}[X]$ and two integers r^* and t^* such that $r^* + t^* \leq \deg(g)$ and $\deg(h) < \deg(g)$, outputs three polynomials $r', s', t' \in \mathbb{K}[X]$ such that $\deg(r') \leq r^*$ and the following property holds : for all polynomials $r, s, t \in \mathbb{K}[X]$ such that $r = sg+th$ with $\deg(r) < r^*$ and $0 \leq \deg(t) \leq t^*$, there exists a non-zero polynomial $u \in \mathbb{K}[X]$ such that $r = r'u$, $s = s'u$ and $t = t'u$.

Question-9. Between the two algorithms of Questions 5 and 8, which one is the most efficient ?