CRYPTOGRAPHIE ALGÉBRIQUE **4M035** **Feuille 1**
P. V. Koseleff

Algorithmique de base

Exercice 1.1. — Montrer que 59 est inversible dans $\mathbf{Z}/1763\mathbf{Z}$ et donner son inverse.

Exercice 1.2. — Écriture matricielle dans l'algorithme d'Euclide
Soit $a,b \in \mathbf{Z}$. Posons $r_0 = a$, $r_1 = b$, $u_0 = 1$, $v_0 = 1$, $u_1 = 1$, v_1 et définissons par récurrence pour $i \geq 1$,

$$r_{i+1} = r_{i-1} - q_i r_i, \quad u_{i+1} = u_{i-1} - q_i u_i, \quad v_{i+1} = v_{i-1} - q_i v_i,$$

où q_i est un entier relatif quelconque.

1. Montrer que pour tout $i \geq 0$, on a $\text{pgcd}(r_i, r_{i+1}) = \text{pgcd}(a,b)$ et $u_i a + v_i b = r_i$.
2. (a) Montrer qu'on a pour tout $i \geq 1$, $\begin{pmatrix} r_i & u_i & v_i \\ r_{i+1} & u_{i+1} & v_{i+1} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & -q_i \end{pmatrix} \cdot \begin{pmatrix} r_{i-1} & u_{i-1} & v_{i-1} \\ r_i & u_i & v_i \end{pmatrix}$
   (b) Montrer que pour tout $i \geq 0$, on a $\begin{vmatrix} u_i & v_i \\ u_{i+1} & v_{i+1} \end{vmatrix} = (-1)^i$.
3. A présent, on définit q_i et $r_{i+1}, u_{i+1}, v_{i+1}$ de la façon suivante, pour $i \geq 1$:
   Si $r_i = 0$ alors $r_{i+1} = r_i$, $u_{i+1} = u_i$, $v_{i+1} = v_i$, sinon $q_i = u_{i-1} \div u_i$ (le quotient de u_{i-1} par u_i, tel que $0 \leq u_{i-1} - q_i u_i < |u_i|$) et $r_{i+1} = r_{i-1} - q_i r_i, \quad u_{i+1} = u_{i-1} - q_i u_i, \quad v_{i+1} = v_{i-1} - q_i v_i$.
   (a) Montrer que la suite r_i est décroissante pour $i \geq 1$.
   (b) Montrer que il existe un rang n tel que $r_n = d \neq 0$ et $r_{n+1} = 0$. Montrer que d est le pgcd de a et b.
   (c) $u_{n+1} = \dfrac{b}{d}(-1)^{n+1}$, $v_{n+1} = \dfrac{a}{d}(-1)^n$.
4. On suppose ici que $0 < b < a$.
   (a) Montrer que les suites u_i et v_i sont de signes alternés et croissantes en valeur absolue.
   (b) Montrer que $|u_n| \leq \dfrac{b}{2d}$, $|v_n| \leq \dfrac{a}{2d}$ (*on notera que* $q_n \geq 2$).

Exercice 1.3. — Théorème Chinois

1. Trouver tous les x vérifiant $x \equiv 98 \pmod{151}, x \equiv 9 \pmod{15}$.
   On veillera dans ce calcul à travailler avec des entiers dans l'intervalle $[0, 15 \cdot 151[$.
2. Soit n et m des entiers premiers-entre-eux. On s'intéresse à la résolution du système $\mathcal{S} = \{x \equiv a \pmod{n}, x \equiv b \pmod{m}\}$.
   (a) Montrer qu'il existe $(u,v) \in \mathbf{Z}^2$ tels que $|u| < |m|$ et $|v| < |n|$ et $un + vm = 1$.
   (b) Soit (u,v) vérifiant la condition précédente. Supposons que $0 \leq a < n, 0 \leq b < m$.
   Montrer que $x = n\Big(u(b-a) \pmod{m}\Big) + a$ est élément de $\mathcal{S}$.
3. Soit $n_1, \ldots, n_k$ des entiers premiers entre-eux deux-à-deux et $n = n_1 \cdots n_k$. Notons $N_i = n/n_i$. On cherche à résoudre le système de congruences $\mathcal{S} = \{x \equiv x_i \pmod{n_i}, i = 1, \ldots, k\}$.
   (a) Montrer qu'il existe u_i et v_i tels que $u_i N_i + v_i n_i = 1$.
   (b) Montrer que $x = \sum_{i=1}^{k} x_i u_i N_i$ est solution de de $\mathcal{S}$.
   (c) Trouver le plus petit x entier tel que $x \equiv 10 \mod 11$, $x \equiv 8 \mod 13$ et $x \equiv 13 \mod 15$.

Exercice 1.4. — Pour $b \in \mathbf{Z}$, on note $\varphi(b)$ le nombre de chiffres en base 2 de $|b|$.

1. Montrer que $\varphi(b) = [\log_2 |b|] + 1$.
2. Montrer que φ est un algorithme euclidien (stathme) : si $(a,b) \in \mathbf{Z} \times \mathbf{Z}^*$, il existe $(q,r) \in \mathbf{Z} \times \mathbf{N}$, tel que $a = bq + r$ et $\varphi(r) < \varphi(b)$.
3. Montrer que φ est le plus petit algorithme euclidien sur $\mathbf{Z}$.

Complexité

On note $\text{len}(a) = [\log_2 a] + 1$ la taille de a. Pour l'instant on admettra (et on démontrera au fil de la séance) que :

1. On peut multiplier deux entiers de tailles n et m en $O(nm)$ opérations binaires (temps de calcul)
2. On peut effectuer la division de a de taille n par b de taille m en $O(m(n-m))$ opérations binaires.
3. On peut calculer le pgcd de deux entiers de taille n en $O(n^2)$ opérations binaires.

Exercice 1.5. — Montrer qu'on peut calculer l'écriture binaire de n en $O((\log n)^2)$ opérations. Idem pour tout changement de base de numération.

Exercice 1.6. — Soit $n_1, \ldots, n_k > 1$ des entiers premiers entre-eux, deux à deux.

1. Montrer qu'on peut calculer $n = \prod_i n_i$ en $O((\text{len}\, n)^2)$ opérations binaires.
2. Montrer que si $a < n$, on peut calculer $(a \bmod n_1, \ldots, a \bmod n_k)$ en $O((\text{len}\, n)^2)$ opérations binaires.
3. Montrer qu'on peut calculer $(n/n_1, \ldots, n/n_i)$ en $O((\text{len}\, n)^2)$ opérations.
4. Montrer qu'on peut calculer k relations de Bézout $u_i N_i + v_i n_i = 1$ en un total de $O((\text{len}\, n)^2)$ opérations.
5. En déduire un majorant du coût de la résolution du système de congruences $\{x \equiv x_i \bmod n_i, i = 1, \ldots, k\}$.

Exercice 1.7. —

1. Combien d'opérations élémentaires faut-il effectuer pour calculer x^{15} par la méthode d'exponentiation rapide ?
2. Peut-on faire mieux ?
3. La complexité de l'exponentiation rapide est $O(\log(n))$. Est-elle optimale ?

Exercice 1.8. — Exponentiation rapide. Soit $n = \sum_{i=0}^{k} \alpha_i 2^i$. On calcule a^n en considérant $\prod_{\alpha_i \neq 0} a^{2^i}$.

1. Ecrire un algorithme qui réalise cette exponentiation rapide.
2. Montrer que qu'on peut calculer a^n en $O(\text{len}\, n)$ opérations arithmétiques dans l'anneau A.
3. Montrer que qu'on peut calculer $a^n \pmod N$ en $O(\text{len}\, n (\text{len}\, N)^2)$ opérations binaires si $a < N$.
4. Montrer que qu'on peut calculer $a^n \in \mathbf{Z}$ en $O((n \text{len}\, a)^2)$ opérations binaires.
5. Comparer le calcul de a^n dans $\mathbf{Z}$ par cet algorithme et par l'algorithme de Horner.
6. Évaluer le calcul de $(X+1)^n$ dans $\mathbf{Z}[X]$ et donc le calcul des coefficients binomiaux $\binom{n}{i}$.

Exercice 1.9. — Suite de Fibonacci.
On considère la *suite de Fibonacci* $(F_n)_{n \geq 0}$ d'éléments de $\mathbf{N}$, définie par $F_0 = 0,\ F_1 = 1,\ F_{n+1} = F_n + F_{n-1}$. On pose $A = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$.

1. Montrer que pour tout entier $n \geq 1$, on a l'identité $A^n = \begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix}$.
2. En déduire la relation $F_{n+1} F_{n-1} - F_n^2 = (-1)^n$.
3. Montrer que $F_{n+m} = F_{n+1} F_m + F_n F_{m-1}$.
4. Montrer que $(F_n, F_{m+n}) = (F_n, F_m)$, puis que $(F_n, F_m) = F_{(n,m)}$.
5. Estimer la complexité du calcul de F_n. Et de $F_n \pmod p$.