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.

Solution 1.1. — 59 est inversible modulo 1763 si et seulement si il est premier avec 1763. Effectuons l'algorithme d'Euclide étendu :

$$\begin{array}{rrcl}
(L_1) & 1763 & = & 1\times 1763+0\times 59,\\
(L_2) & 59 & = & 0\times 1763+1\times 59,\\
(L_3=L_1-29L_2) & 52 & = & 1\times 1763-29\times 59,\\
(L_4=L_2-L_3) & 7 & = & -1\times 1763+30\times 59,\\
(L_5=L_4-7L_3) & 3 & = & 8\times 1763-239\times 59,\\
(L_6=L_4-2L_5) & 1 & = & -17\times 1763+508\times 59,
\end{array}$$

On obtient la relation de Bézout $1=-17\times 1763+508\times 59$. L'inverse de 59 est 508, puisque $508\times 59\equiv 1 \pmod{1763}$.

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 $\operatorname{pgcd}(r_i,r_{i+1})=\operatorname{pgcd}(a,b)$ 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* $q_n\geq 2$).

Solution 1.2. —

1. On a en effet $\begin{pmatrix} u_i & v_i\\ u_{i+1} & v_{i+1}\end{pmatrix}=\begin{pmatrix}0 & 1\\ 1 & -q_i\end{pmatrix}\cdot\begin{pmatrix} u_{i-1} & v_{i-1}\\ u_i & v_i\end{pmatrix}$, d'où le résultat en considérant le déterminant.
2. Une réponse fausse fréquente est d'utiliser l'unicité de la solution de l'équation $ax+by=d[ab]$ car on n'a pas d'estimation sur $|u_{n+1}|$ et $|v_{n+1}|$. La solution correcte consiste à utilise 2.(a) et a écrire : $\begin{vmatrix} r_{i-1} & u_{i-1}\\ r_i & u_i\end{vmatrix}=(-1)^{i-1}\begin{vmatrix} r_0 & u_0\\ r_1 & u_1\end{vmatrix}$, soit lorsque $i=n+1$: $du_{n+1}=(-1)^n\cdot(-b)$. On fait de même avec v_{n+1}.
3. De $u_{k+1}=u_{k-1}-q_k u_k$, on déduit que la suite u_k est de signe alterné, pour $k\geq 1$, puis $|u_{k+1}|=|u_{k-1}|+|q_k u_k|>|u_k|$. Mais $q_n\neq 1$ car $r_{n-1}=q_n r_n>r_n$ donc $q_n\geq 2$ et $|u_n|\leq\frac{1}{2}|u_{n+1}|$.

L'algorithme d'Euclide étendu fournit donc une identité de Bézout et une solution de l'équation homogène (dans $\mathbf{Z}^2$) : $ax+by=0$. Il est remarquable, que la solution obtenue par l'algorithme d'Euclide étendu engendre l'ensemble des solutions de l'équation homogène et que la solution particulière soit une solution particulière minimale.

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 $x \equiv 13 \mod 15$.

Solution 1.3. —

1. On écrit tout d'abord une relation de Bézout : $1 \cdot 151 - 10 \cdot 15 = 1$ d'où on déduit que le nombre $x = 151b - 150a$ vérifie la congruence $x \equiv a \pmod{151}, x \equiv b \pmod{15}$. On obtient alors ici $x = 9 * 151 - 98 * 150 = -13341$ ce qui est trop gros et qu'il faut réduire modulo $15 * 151 = 2265$. On obtient $x = 249$. Reprenons le calcul, et écrivons $x = ((9-98) \pmod{1)5} * 151 + 98 = (-89 \pmod{1)5} * 151 + 98 = 151 + 98 = 249$.
2. (a) Soit (u_0, v_0) une solution de $un+vm = 1$, obtenue par l'algorithme d'Euclide étendu. Tout autre solution s'écrit $(u,v) = (u_0, v_0) + k(m, -n)$, où $k \in \mathbf{Z}$. Choisisons k, tel que $0 \leq v_1 = v_0 - kn < n$. On a alors, en posant $u_1 = u_0 + km$, $|u_1 n| = |1 - v_1 m| \leq 1 + |v_1 m| \leq 1 + (n-1)m < nm$. Alors $|u_1| < m$.
   (b) Soit $x = n\Big(u(b-a) \pmod{m}\Big) + a$. On a $|u(b-a)| \leq |u||b-a| \leq (m-1)n < nm$. Puis $0 \leq \Big(u(b-a) \pmod{m}\Big) \leq m-1$ et enfin $n\Big(u(b-a) \pmod{m}\Big) \leq n(m-1)$, soit $x \leq n(m-1) + (n-1) = nm - 1$.
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 $u_i N_i + v_i n_i = 1$.*
   $(n_i, N_i) = 1$ donc on applique Bézout.
   (b) *Montrer que $x = \sum_{i=1}^{k} x_i u_i N_i$ est solution de de $\mathcal{S}$.*
   x vérifie bien les congruences puisque $u_i N_i \equiv \delta_{i,j} \pmod{n_j}$.
   (c) On obtient $x = 365 * 10 + 495 * 8 + 286 * 13 \pmod{13 * 33 * 35} = 2023$.

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 $\mathbf{Z}$.

Solution 1.4. — Le fait que φ soit un algorithme euclidien figure est clair. Soit ψ un autre algorithme euclidien sur $\mathbf{Z}$ à valeurs dans $\mathbf{N}$: on va démontrer par récurrence sur $n = \psi(b)$ que $\varphi(b) \leq \psi(b)$.

C'est vrai pour $n = 0$ car $\psi(b) = 0$ entraîne $b = 0$; supposons que ce soit vrai pour tous les b tels que $\psi(b) < n$ et démontrons le pour b tel que $\psi(b) = n$.

Effectuons la division ψ-euclidienne de $\lfloor |b|/2 \rfloor$ par b. On a $\lfloor |b|/2\ rfloor = bq + r$, avec $\psi(r) < \psi(b)$. Puisque r est un représentant de $\lfloor |b|/2 \rfloor$ modulo b, alors $|r| \geq \lfloor |b|/2 \rfloor$, d'où l'on tire :

$$\varphi(r) \geq \varphi(\lfloor |b|/2 \rfloor) = \varphi(b) - 1$$

Puisque $\psi(r) < n$, par récurrence on obtient $\psi(r) = \varphi(r)$, et par conséquent :

$$\varphi(b) \leq \varphi(r) + 1 = \psi(r) + 1 \leq \psi(b)$$

ce qui termine la démonstration.

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.

Solution 1.5. — Si $n = \sum_{i=0}^{k} \alpha_i 2^i$ alors on a $n = 2q_0 + \alpha_0$. On calcule donc successivement $q_0 = n$, $\alpha_0 = 0$, puis $\alpha_i = q_i \bmod 2$, $q_{i+1} = q_i \div 2$, tant que $q_i > 0$. On a $q_i \leq n/2^i$.

On a donc $\lfloor \log_2 n \rfloor$ étapes. On calcule le couple (α_i, q_{i+1}) en $O(\log q_i) = O(\log n)$ opérations, soit au total $O((\log n)^2)$ opérations. En fait, on peut améliorer car

$$\sum_{i=0}^{k} \log q_i \leq \sum_{i=0}^{k} (\log n - i) \sim \tfrac{1}{2}(\log n)^2$$

Cela ne change pas l'ordre de grandeur.

Le principe reste le même si on utilise B plutôt que 2.

Mais n est donné sous quelle forme ? Supposons que n soit donné en base 2 ? On commence par écrire B en base 2, puis on utilise l'algorithme précédent où B et q_i sont écrits dans la même base.

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 ?

Solution 1.7. —

1. 15 s'écrit 1111 en binaire. Il faut donc effectuer 6 multiplications pour calculer successivement x^2, x^4 et x^8, puis rassembler les morceaux pour obtenir x^{12}, x^{14} et enfin x^{15}.
2. On peut se contenter de 5 multiplications pour calculer successivement x^2, x^4, x^5, x^{10} et x^{15}.
3. Par récurrence sur r, on montre que le plus grand m tel que x^m est calculable en r multiplications est $m = 2^r$. Si $r = 1$, on peut calculer x^2. Si $r = 3$, on peut calculer $[x^2, x^3]$ ou $[x^2, x^4]$, donc $m = 4 = 2^r$. Supposons le résultat vrai pour $r \geq 2$. Supposons qu'on puisse calculer $x^{a_1}, \ldots, x^{a_{r+1}}$ en $r+1$ multiplications où $a_{r+1} > 2^{r+1}$. On a alors $a_{r+1} = a_i + a_j$ où $a_i \geq a_j \geq 1/2(a_i + a_j) = 1/2a_{r+1}$. Mais alors $a_i > 2^r$ et x^{a_i} se calcule en au moins $r+1$ multiplications. Alors $x^{a_{r+1}}$ se calcule au moins en $r+2$ multiplications. Donc $a_{r+1} \leq 2^{r+1}$. On en déduit que le nombre d'opérations pour calculer x^n est au moins égal à $\log_2(n)$. La complexité $O(\log(n))$ est donc 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 $\binom{n}{i}$.

Solution 1.8. —

1. Considérons l'algorithme suivant :

   Algorithme 1.1. — *Exponentiation rapide*

Entrées : $a \in A$, $n = \sum_{i=0}^{k} n_i 2^i$.
Sorties : a^n.

(a) $r := 1$, $\mathbf{a} = a$ # $\mathbf{a} = a$ *contient* a^{2^i}
(b) Pour i de 0 à $k-1$
   i. Si $n_i = 1$, $r := r \cdot \mathbf{a}$
   ii. $\mathbf{a} := \mathbf{a}^2$
(c) Si $n_i = 1$, $r := r \cdot \mathbf{a}$

   À chaque moment on a $r = a^{\sum_{j=0}^{i} n_j}$ et on a bien à la fin $r = a^n$.
2. $k = \text{len}\, n$. On effectue $2(k-1) = O(\log n)$ multiplications (élévation au carré $\mathbf{a} := \mathbf{a}^2$ et produit $r := r \cdot \mathbf{a}$).
3. Tous les a^i sont de taille bornée par $\text{len}\, N$. On effecteue $O(\text{len}\, n)$ multiplications entre éléments de taille $(\text{len}\, N)$, chacune coûtant $O((\text{len}\, N)^2)$ opérations binaires. Le coût total est donc $O(\text{len}\, n(\text{len}\, N)^2)$.
4. a^k est de taille $k \log a$. Le produit de a^k par a^i se fait en $O(ik(\log a)^2)$. En utilisant l'exponentiation rapide, à chaque étape, on multiplie (ou pas) a^{2^i} par a^j où $j < 2^i$. On peut majorer le coût par $\sum_i O(4^i(\text{len}\, a)^2) = O((n \log a)^2)$. Dans le cas où $n = 2^k - 1$, les majorations précédentes sont fines.
   Dans le cas de la méthode de Horner, on multiplie (ou pas) à chaque étape $r = a^i$ par a en $O(i(\log a)^2)$, soit en sommant, au total en $O((n \log a)^2)$ opérations élémentaires.
   En conclusion, l'exponentiation rapide n'est pas plus rapide, ici.
5. Si on utilise le schéma de Horner, pour calculer $(X+1)^n$, on calcule la suite $H_0 = 1$, $H_{i+1} = H_i(X+1) = XH_i + H_i$. Le coût de $X \cdot H_i$ est (presque)nul (il s'agit de décaler les puissances). Le coût de $XH_i + H_i$ est de $O(i)$ opérations arithmétiques (H_i est de degré i). Il s'agit d'additionner $i+1$ coefficients de taille $O(i)$. On calcule donc H_{i+1} en $O(i^2)$ opérations binaires à partir de H_i. Au final, on calcule H_n en $O(n^3)$ opérations binaires.

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 $F_{n+m} = F_{n+1}F_m + F_nF_{m-1}$.
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 $F_n \pmod p$.