Feuille 2 : Complexité, arithmétique et algorithme d'Euclide

Exercice 1. Soient des entiers $m \geq n \geq 1$ et $a \geq 2$.

1. Soit r le reste de la division euclidienne de m par n. Montrer que le reste de la division de $a^m - 1$ par $a^n - 1$ est $a^r - 1$.
2. En déduire que $\text{pgcd}(a^m - 1, a^n - 1) = a^{\text{pgcd}(m,n)} - 1$.
3. Retrouver ce résultat en montrant que si un entier d divise $a^m - 1$ et $a^n - 1$, alors d divise $a^{\text{pgcd}(m,n)} - 1$.

Exercice 2.

1. Trouver le pgcd et une relation de Bézout pour les couples (a, b) suivants : $(136, 51)$ et $(57, 13)$.
2. Calculer l'inverse de 25 dans l'anneau $\mathbf{Z}/37\mathbf{Z}$.
3. Donner la liste des éléments x inversibles de l'anneau $\mathbf{Z}/12\mathbf{Z}$ et de leurs inverses x^{-1}. Même question dans $\mathbf{Z}/9\mathbf{Z}$.

Exercice 3.

1. Trouver tous les entiers n vérifiant $2n \equiv 5 \pmod{21}$.
2. Déterminer tous les $x \in \mathbf{Z}/25\mathbf{Z}$ vérifiant $20x = 1$. Même question pour l'équation $20x = 10$.
3. Trouver tous les couples $(x, y) \in \mathbf{Z}^2$ vérifiant :

$$x + y \equiv 6 \pmod{11} \quad \text{et} \quad 2x - y \equiv 8 \pmod{11}.$$

Exercice 4. Trouver tous les entiers n vérifiant : $2n \equiv 3 \pmod{5}$ et $3n \equiv 1 \pmod{7}$.

Exercice 5. Trouver tous les entiers n vérifiant

$$\begin{cases} 5n \equiv 6 \pmod{12} \\ 2n \equiv 3 \pmod{15} \\ 2n \equiv 4 \pmod{7} \end{cases}$$

Exercice 6. Algorithme de Karatsuba.

1. Soient a et b deux entiers naturels $< 2^{2n}$ avec n un entier ≥ 1. Comment calculer ab en n'effectuant que trois multiplications d'entiers naturels $\approx 2^n$? Indication : écrire $a = 2^n a_1 + a_2$ et $b = 2^n b_1 + b_2$ puis considérer $(a_1 + a_2)(b_1 + b_2)$.
2. Soient a et b deux entiers naturels $< 2^n$ avec $n = 2^m$ pour un entier m. Proposer un algorithme calculant ab de complexité $\mathcal{O}(n^\alpha)$ avec $\alpha = \log_2(3) \approx 1.58$.

Exercice 7. Soit N un entier naturel non nul.

1. Soit $n = \log_2 N$ et $n' = \log_e N$. Montrer qu'un algorithme en $\mathcal{O}(n)$ est aussi en $\mathcal{O}(n')$ et réciproquement.
2. Quelle est la complexité de l'addition modulo N, c'est à dire l'algorithme qui prend en entrée (a, b) avec $0 \leq a < N$, $0 \leq b < N$ et retourne c tel que $0 \leq c < N$ et $c \equiv a + b \pmod N$? Quelle est la complexité de la multiplication modulo N ?
3. Le chiffrement affine envoie $m \in \mathbf{Z}/N\mathbf{Z}$ sur $am + b$ où $a, b \in \mathbf{Z}/N\mathbf{Z}$. Quelle est la complexité de cet algorithme de chiffrement ?
4. Le chiffrement de Hill envoie $m \in (\mathbf{Z}/N\mathbf{Z})^\ell$ sur mK où $K \in GL_\ell(\mathbf{Z}/N\mathbf{Z})$ avec ℓ un entier naturel non nul. Quelle est celle de cet algorithme de chiffrement ?
5. Le chiffrement par décalage envoie $m \in \mathbf{Z}/N\mathbf{Z}$ sur $m + k$ où $k \in \mathbf{Z}/N\mathbf{Z}$. On dispose d'un couple (m, c) message clair, message chiffré correspondant, pour ce chiffrement. On souhaite retrouver la clef de chiffrement. Quelle est la complexité dans le pire cas de l'attaque naïve par recherche exhaustive ? Quelle est celle de l'attaque « intelligente » ?

Exercice 8. On se place dans $\mathbf{Z}/23\mathbf{Z}$.

1. Calculer 2^7 et 3^8 par l'algorithme d'exponentiation modulaire.
2. Combien avez-vous effectué de multiplications modulaires dans chacun des cas ? Dans le cas général, combien faut il de multiplications modulaires pour calculer a^k dans $\mathbf{Z}/N\mathbf{Z}$ avec k un entier non nul de ℓ bits et N un entier naturel non nul.

Exercice 9. Complexité de l'algorithme d'Euclide

Si x est un entier, on note $L(x) = E(\log_2(x)) + 1$ le nombre de bits de x. Soient a et b deux entiers tels que $a > b > 0$. On note $r_0 = a$ et $r_1 = b$. On applique l'algorithme d'Euclide à a et b de la manière suivante :

$$\begin{aligned} r_0 &= r_1 q_1 + r_2 \\ r_1 &= r_2 q_2 + r_3 \\ &\vdots \\ r_{n-2} &= r_{n-1} q_{n-1} + r_n \\ r_{n-1} &= r_n q_n \end{aligned}$$

où $r_n = \text{pgcd}(a, b)$.

1. La suite des nombres de Fibonacci est définie par $F_0 = 0, F_1 = 1$, $F_i = F_{i-1} + F_{i-2}$ pour $i \geq 2$. Calculer $\text{pgcd}(F_{n+2}, F_{n+1})$ pour $n \geq 0$. Combien fait-on d'étapes et que valent les quotients ?
2. Montrer que si le calcul de $\text{pgcd}(a, b)$ utilise n divisions euclidiennes alors $b \geq F_{n+1}$. *Indication* : remonter l'algorithme.
3. Vérifier que pour tout $n \geq 2$, $F_n \geq \Phi^{n-2}$ où $\Phi = \frac{1+\sqrt{5}}{2}$ est le nombre d'or. En déduire que l'algorithme d'Euclide appliqué à a et b s'arrête après $\mathcal{O}(L(b))$ divisions euclidiennes (Théorème de Lamé).
4. Montrer que dans l'algorithme d'Euclide le produit des quotients est majoré par a, c'est-à-dire que $q_1 q_2 \dots q_n \leq a$.
5. La division euclidienne de x par y peut se faire en $\mathcal{O}(L(y)L(q))$ opérations binaires où q est le quotient. Montrer que la complexité de l'algorithme d'Euclide est en $\mathcal{O}(L(a)L(b))$

Exercice 10. Inverser par exponentiation

1. Utiliser le fait que pour tout x de $(\mathbf{Z}/n\mathbf{Z})^\times$ on a $x^{\varphi(n)} = 1$ pour proposer une façon de calculer l'inverse de x dans $(\mathbf{Z}/n\mathbf{Z})^\times$, qui ne fasse pas appel à l'algorithme d'Euclide étendu. Quelle est la complexité de cet algorithme ?
2. Application : calcul de l'inverse de 7 dans $(\mathbf{Z}/15\mathbf{Z})^\times$.