SÉANCE 4. ALGORITHME D'EUCLIDE ET CONGRUENCES

Objectifs : écriture matricielle de l'algorithme d'Euclide, exponentiation rapide, théorème chinois et calcul de la fonction indicatrice d'Euler

Exercice 1 (Algorithme d'Euclide) — Soit $a, b \in \mathbf{N}$ deux nombres entiers.

1. Vérifier les identités matricielles
$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} b \\ a \end{pmatrix} \text{ et } \begin{pmatrix} 1 & -q \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} a - qb \\ b \end{pmatrix}$$
pour tout $q \in \mathbf{Z}$.
2. En déduire un algorithme `Euclide(a,b)` calculant $d = \text{pgcd}(a, b)$ et une matrice P de taille 2×2 à coefficients entiers telle que $\det(P) = \pm 1$ et
$$P \cdot \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} d \\ 0 \end{pmatrix}.$$
3. Que peut-on dire des coefficients de la première ligne de P ?

Exercice 2 (Exponentiation rapide) — Soit $N \geqslant 2$ et $n \geqslant 1$ deux nombres entiers. Étant donné un nombre entier $a \in \mathbf{Z}$, nous allons voir que le calcul de a^n modulo N peut s'effectuer très rapidement.

1. Soit $N \geqslant 2$, $n \geqslant 0$ et $a \in \mathbf{Z}$ des nombres entiers. Écrire un algorithme `ExpNaive(a,n,N)` qui calcule a^n modulo N en effectuant $n-1$ multiplication dans $\mathbf{Z}/N\mathbf{Z}$.
2. Fixons un nombre entier $b \geqslant 2$.
   (i) Démontrer que tout nombre entier $n \geqslant 0$ s'écrit d'une manière et d'une seule sous la forme
   $$n = x_0 b^0 + x_1 b^1 + \ldots + x_k b^k$$
   avec $x_0, \ldots, x_k \in \{0, \ldots, b-1\}$ et $x_k \neq 0$. Les x_i sont les *chiffres* de n en base b.
   (ii) Vérifier en outre que l'on a
   $$k = \left[\frac{\log n}{\log b} \right].$$
   (iii) Écrire un algorithme `Base(n,b)` qui renvoie la liste $[x_0, x_1, \ldots, x_k]$ des chiffres de n en base b. Combien de division euclidienne utilisez-vous ?
3. Notons $\varepsilon_0, \ldots, \varepsilon_k$ les chiffres de n en base 2. Pour tout entier $a \in \mathbf{Z}$,
$$a^n = a^{\varepsilon_0 + 2\varepsilon_1 + \ldots + 2^k \varepsilon_k} = \prod_{i=0}^{k} \left(a^{2^i}\right)^{\varepsilon_i} = \prod_{0 \leqslant i \leqslant k,\ \varepsilon_i \neq 0} a^{2^i}.$$
Pour calculer a^n modulo N, il est donc suffisant de calculer les carrés itérés
$$a^2, a^4 = (a^2)^2, \ldots, a^{2^k} = (a^{2^{k-1}})^2$$
modulo N, puis de faire le produit (modulo N) des a^{2^i} correspondant aux indices i tels que $\varepsilon_i \neq 0$.

(i) Écrire un algorithme `ExpRapide(a,n,N)` qui calcule a^n modulo N de la façon que l'on vient de décrire.

(ii) Combien d'opérations (divisions euclidiennes et multiplications) cet algorithme requiert-il ?

(iii) Comparer le temps de calcul des algorithmes `ExpNaive`(a,n,N) et `ExpRapide`(a,n,N).

4. Déterminer les quatre dernières décimales de 12^{1000}.

5. On rappelle que le n-ième nombre de Fermat est défini par $F_n = 2^{2^n} + 1$.

   (i) Calculer 3^{F_5-1} modulo F_5.

   (ii) En déduire que F_5 n'est pas premier.

Exercice 3 (Théorème des restes chinois) — Rappelons le *théorème des restes chinois* :

étant donné un nombre entier $m > 1$ et une factorisation $m = m_1 m_2 \cdots m_r$ en produit d'entiers $m_i > 1$ deux à deux premiers entre eux*, l'application naturelle*

$$f : \mathbf{Z}/m\mathbf{Z} \longrightarrow \mathbf{Z}/m_1\mathbf{Z} \times \mathbf{Z}/m_2\mathbf{Z} \times \cdots \times \mathbf{Z}/m_r\mathbf{Z}$$

$$x \bmod m \longmapsto (x \bmod m_1, x \bmod m_2, \ldots, x \bmod m_r)$$

est un isomorphisme d'anneaux.

1. Soit L une liste de nombres entiers. Écrire un algorithme `Test(L)` vérifiant si les coefficients de L sont deux à deux premiers entre eux.

2. Si $L = [m_1, \ldots, m_r]$ est une liste d'entiers premiers deux à deux premiers entre eux, écrire un algorithme `Chinois(a,L)` calculant, pour tout $a \in \mathbf{Z}$, les composantes de $f(a)$.

3. Considérons toujours une liste $L = [m_1, \ldots, m_r]$ d'entiers premiers entre eux deux à deux et posons $m = m_1 m_2 \cdots m_r$.

   (i) Soit $i_0 \in \{1, \ldots, r\}$. À l'aide d'une identité de Bézout bien choisie, déterminer un entier e_{i_0} tel que

   $$e_{i_0} \equiv 1 \bmod m_{i_0} \quad \text{et} \quad e_{i_0} \equiv 0 \bmod m_i$$

   pour tout $i \neq i_0$.

   (ii) En déduire un algorithme `ChinoisInverse(a,L)` associant à toute liste $[a_1, a_2, \ldots, a_r]$ de nombres entiers l'unique entier $a \in \{0, \ldots, m-1\}$ tel que $a \equiv a_i \bmod m_i$ pour tout i.

4. En guise d'application, considérons $m = 100 \cdot 101 \cdot 103$. Déterminer un nombre entier a tel que $a \equiv 60 \bmod 103$ et $\operatorname{pgcd}(a, m) = 1$.

Exercice 4 (La fonction indicatrice d'Euler) — On rappelle que l'indicatrice d'Euler est la fonction φ qui a un entier $n > 0$ associe le nombre d'entiers m compris entre 1 et n et premiers à n.

1. Montrer que φ est une fonction *multiplicative*, c'est-à-dire qu'étant donnés $n, m \geq 1$ premiers entre eux, on a $\varphi(nm) = \varphi(n)\varphi(m)$.

   Indication : Remarquez que pour tout entier $N \geq 1$, $\varphi(N)$ est le cardinal de l'ensemble $(\mathbf{Z}/N\mathbf{Z})^*$ des éléments inversibles de $\mathbf{Z}/N\mathbf{Z}$. Utilisez ensuite le théorème des restes chinois.

2. Soit p un nombre premier et $k \geq 1$ un entier. Calculez $\varphi(p)$ puis justifiez que $\varphi(p^k) = (p-1)p^{k-1}$. Soit n un entier ≥ 1. Déduire une expression de $\varphi(n)/n$ en fonction des facteurs premiers de n.

3. Ecrire une fonction `ListeEntiersPremiers(n)` renvoyant la liste des entiers m compris entre 1 et n et premiers à n.
4. Ecrire une fonction `Phi(n)` renvoyant $\varphi(n)$.
5. Ecrire une version récursive `PhiRecursive(n)` de la fonction précédente. On pourra penser à d'abord écrire une fonction intermédiaire `DecompoPartielle(n)` prenant en argument n et renvoyant (p,k,m), avec p,k,m des entiers ≥ 1 tels que $n = p^k m$ et p ne divisant pas m.
6. Ecrire une fonction `SumPhi(n)` renvoyant $\sum_{d|n} \varphi(d)$. Que peut-on conjecturer sur `SumPhi` ?
7. Soit n un entier ≥ 1. Pour tout diviseur d de n, on note

$$A_d := \{1 \leq k \leq n; k \wedge n = d\}.$$

   Montrez que A_d est l'ensemble des k s'écrivant $d \times \ell$ avec ℓ un entier compris entre 1 et n/d premier à n/d. En déduire le cardinal de A_d, puis que

$$\sum_{d|n} \varphi(d) = n.$$

8. En utilisant la formule de la question précédente, écrire une version récursive `PhiRecursive2(n)` calculant $\varphi(n)$.
9. Tracez l'ensemble des $\varphi(n)/n$ pour n compris entre 1 et 100.
10. Trouvez une suite $(a_n)_{n\geq 1}$ d'entiers telle que $\varphi(a_n)/a_n$ tende vers 1 lorsque n tend vers l'infini.
11. (Difficile) On note $(p_i)_{i\geq 1}$ la suite des nombres premiers ordonnés par ordre croissant et on définit $b_k = p_1 \cdots p_k$ pour tout $k \in \mathbf{N}$. Montrez que $\varphi(b_n)/b_n$ tend vers 0 lorsque n tend vers l'infini.

    Indication : En remarquant que $1/(1-1/p) = \sum_{k\geq 0} 1/p^k$, justifiez que

$$\prod_{p \text{ premiers}} \frac{1}{1-1/p} = \sum_{n\geq 1} \frac{1}{n} = +\infty.$$