CMA : Devoir à la maison 2005

Rendre les copies à votre enseignant de groupe avant la fin de novembre

Algorithme d'Euclide, théorème des restes chinois, interpollation polynomiale

Algorithme d'Euclide cherche le plus grand commun diviseur de deux entiers a et b par des divisions successives. Soit d le pgcd de a et b, $a > b$; alors, pour trouver d en fonction de a et b on fait les divisions suivantes :

$$\begin{aligned} a &= q_0 b + r_1 \\ b &= q_1 r_1 + r_2 \\ r_1 &= q_2 r_2 + r_3 \\ \dots &= \dots \\ r_{k-1} &= q_k r_k \,. \end{aligned}$$

Quand on obtient $r_{k+1} = 0$, on pose $d = r_k$.

Exercice 1. (a) Appliquer l'algorithme d'Euclide à $a = 182$, $b = 49$.

(b) Appliquer l'algorithme d'Euclide à deux nombres de Fibonacci consécutifs 55 et 34, puis à 89 et 55, puis à 144 et 89.

(c) Montrer que pour n fixé, le maximum du nombre des divisions dans l'algorithme d'Euclide sur toutes les paires (a, b), $b < a \leq n$ est atteint sur une paire de nombres de Fibonacci consécutifs.

(d) On sait que la croissance des nombres de Fibonacci est exponentielle : $f_k \approx \varphi^k$ où $\varphi = (1 + \sqrt{5})/2 \approx 1.618$. Montrer que si $b < a \leq n$ alors le nombre des divisions dans l'algorithme d'Euclide est $O(\log n)$.

Si quelqu'un nous donne d en prétendant que $d = \text{pgcd}(a, b)$, sommes-nous en mesure de vérifier que cette affirmation soit correcte ? Nous povons bien, en faisant seulement deux divisions (a/d et b/d), vérifier que d soit un commun diviseur. Mais comment vérifier qu'il soit *le plus grand* divisuer ? Apparemment, il faut réappliquer l'algorithme d'Euclide du début à la fin :-(.

La méthode proposée pour résoudre ce problème est basée sur la propriété suivante : $d = \text{pgcd}(a, b)$ si et seulement si

(1) d divise a et b ;
(2) il existe x et y tels que $ax + by = d$.

Alors si, à part d, on nous communique aussi x et y, nous pouvons vérifier l'affirmation $d = \text{pgcd}(a, b)$ en faisant encore trois opérations arithmétiques pour calculer $ax + by$.

Exercice 2 (Algorithme d'Euclide autocertifié). Modifier, sans changer sa complexité $O(\log n)$, l'algorithme d'Euclide de telle manière qu'il sorte, à part d, aussi x et y tels que $ax + by = d$.

Indication : exprimer r_1 de la première égalité ($a = q_0 b + r_1$) ; exprimer r_2 de la deuxième égalité ; ainsi de suite... ; exprimer $r_k = d$ de la dernière égalité.

Théorème (Théorème des restes chinois). Soit $N = m_1 m_2 \dots m_k$, tous les m_i étant premiers entre eux. Alors pour toute suites de restes $s_1 \bmod m_1$, $s_2 \bmod m_2$, ..., $s_k \bmod m_k$ il existe un et un seul reste $s \bmod N$ tel que $s = s_1 \bmod m_1$, $s = s_2 \bmod m_2$, ..., $s = s_k \bmod m_k$.

Exercice 3. (a) Trouver un reste s mod 12 tel que $s = 1 \bmod 3$ et $s = 2 \bmod 4$.
(b) Trouver un reste s mod 35 tel que $s = 2 \bmod 5$ et $s = 4 \bmod 7$.
(c) Vérifier que les solutions obtenues dand ces deux cas sont uniques.

Exercice 4. Soit $n_i = N/m_i$. Montrer que l'algorithme suivant donne la solution du problème des restes chinois :

pour i de 1 à k faire
  trouver x_i et y_i tels que $x_i n_i + y_i m_i = 1$ (voir l'Exercice 2)
  $c_i = x_i s_i$
fin-pour
retourner $s = \sum_{i=1}^{k} c_i n_i \bmod N$

Estimer la complexité de cet algorithme.

L'opération de la division avec le reste existe pas seulement pour les entiers mais aussi pour les polynômes. Cette simple remarque implique que tous les algorithmes précédents (algorithme d'Euclide, algorithme d'Euclide autocertifié, algorithme des restes chinois) peuvent être transportés, sans aucun changement, sur les polynômes. Seule la complexité change car une opération sur les polynômes (addition, multiplication, division) n'est plus *une* opération arithmétique mais plusieurs.

Exercice 5. Soit $m_0(x) = x - x_0$, $m_1(x) = x - x_1$, ..., $m_{n-1}(x) = x - x_{n-1}$. Montrer que l'apllication de l'algorithme des restes chinois aux polynômes $m_0(x), m_1(x), \ldots, m_{n-1}(x)$ n'est rien d'autre que l'interpollation de Lagrange.

Exercice 6 (Secret partagé). Soit a_0 le code secret d'une carte bleue (4 chiffres décimaux). On veut faire en sorte que n personnes ensemble puissent récupérer ce code mais aucun sous-groupe de $n-1$ personnes ne soit pas en mesure de le faire. Pour cela, on utilise la méthode suivante. On prends un nombre premier p ayant plus que 4 chiffres et supérieur à n ; on choisit par hasard n éléments distincts $x_0, x_1, \ldots, x_{n-1} \bmod p$, puis encore $n-1$ éléments $a_1, a_2, \ldots, a_{n-1} \bmod p$ (rapelons que a_0 est fixé dès le début). Soit $P = P(x)$ le polynôme

$$P(x) \;=\; a_0 + a_1 x + a_2 x^2 + \ldots + a_{n-1} x^{n-1}\,.$$

Chacune des n personnes obtient comme information la valeur $y_i = P(x_i) \bmod p$. Montrer que ces données résolvent bien le problème posé.

Question-bonus (Interpollation d'Hermite). Dans interpollation de Lagrange, on cherche n coefficients d'un polynôme $P(x)$ de degré $n-1$ à partir de n valeurs de ce polynôme dans des points $x_0, x_1, \ldots, x_{n-1}$. Dans l'interpollation d'Hermite, on se donne k points $x_0, x_1, \ldots, x_{k-1}$, et dans chacun de ces points on se donne la valeur $P(x_i)$ et les valeurs de $p_i - 1$ premières dérivées : $P'(x_i), P''(x_i), \ldots, P^{(p_i-1)}(x_i)$, avec la condition que $\sum_{i=0}^{k-1} p_i = n$. On cherche, comme avant, les coefficients du polynôme P.

Montrer que le problème d'interpollation d'Hermite est équivalent au problème des restes chinois pour les polynômes $m_0(x) = (x - x_0)^{p_0}$, $m_1(x) = (x - x_1)^{p_1}$, ..., $m_{k-1}(x) = (x - x_{k-1})^{p_{k-1}}$.

Fin de l'énoncé