Mathématiques pour l'informatique

LSI 1

05 - 05 - 2008

Deux heures, documents et calculatrice interdits.

1 Algorithme d'exponentiation rapide

Le but de l'exercice est l'écriture et l'analyse de l'algorithme dit *d'exponentiation rapide*, ou encore *Square and Multiply*. Cet algorithme permet de calculer rapidement a^n, où a est un élément d'un anneau A quelconque et n un entier naturel. Il n'est pas plus dur à comprendre dans un anneau quelconque, où, rappelons-le, on a défini une multiplication, que dans l'anneau des réels, et on en verra une application utile en cryptographie à la dernière question.

Le principe de cet algorithme est très simple, et vous l'avez certainement déjà utilisé pour le calcul de a^n lorsque a est un simple réel : plutôt que d'effectuer $n-1$ multiplications de a avec lui-même (ce qui constitue ce que l'on appellera dans la suite l'algorithme "naïf"), il vaut mieux utiliser des élévations au carré successives, et compléter le calcul si nécessaire par quelques multiplications (par des termes d'ailleurs déjà calculés).

1. (a) À titre d'exemple, expliquez comment calculer a^{19} avec ce principe, compter le nombre de multiplications que vous avez du faire, et le comparer avec le nombre de multiplications que nécessite l'utilisation de l'algorithme naïf pour ce même calcul (1 *point*).

   (b) Plus généralement, si n s'écrit $\sum_{j=0}^{k} b_j 2^j$ en base 2 (donc les b_j valent 0 ou 1), donner la formule permettant le calcul de a^n en suivant ce principe (1 *point*).

2. On note 1_A l'élément neutre de la multiplication dans l'anneau A. Voici l'algorithme d'exponentiation rapide :

```
Fonction ExponentiationRapide (a : élément de A, n : entier naturel)
Sortie : élément de A ;
Variables   x, Res : éléments de A, N : entier naturel ;
Début
    x := a ; N := n ; Res := 1_A ;
    Tant que N > 0 faire
        Si N mod 2 = 0 alors x := x * x ; N := N div 2 ;
        Sinon Res := Res * x ; N := N - 1 ;
        Finsi
    FinTantque
    Retourne Res ;
Fin
```

   (a) Vérifier que l'algorithme fait bien ce qu'on attend de lui, *i.e* qu'il se termine et qu'en sortie la variable *Res* contient l'élément a^n de l'anneau A (2,5 *points*).

   (b) Calculer la complexité de l'algorithme dans le pire des cas en fonction de la taille de n, et comparer avec la complexité de l'algorithme naïf (2 *points*).

3. On souhaite utiliser l'algorithme dans le cas où l'anneau A est $\mathbb{Z}/p\mathbb{Z}$; autrement dit, on souhaite calculer efficacement $a^n \bmod p$ à l'aide de cet algorithme (ceci est indispensable dans plusieurs méthodes modernes de chiffrement). Quelles adaptations de l'algorithme "brut" proposez-vous pour y parvenir (1,5 *point*) ?

2 Mathématiques du signal

Soit $a > 0$. On note f_a la fonction indicatrice de $[-a, a]$, *i.e.* la fonction valant 1 sur $[-a, a]$ et 0 ailleurs.

1. Soit a et b deux paramètres > 0. Calculer le produit de convolution $f_a * f_b$ (2 *points*).
2. Donner la transformée de Fourier de $f_a * f_b$, et en déduire la valeurs des intégrales

$$I(\omega) = \int_0^{+\infty} \frac{\sin(t)\sin(2t)\cos(\omega t)}{t^2} dt \quad \text{et} \quad J = \int_0^{+\infty} \frac{\sin^2(t)\sin^2(2t)}{t^4} dt$$

(3 *points*)

3 Equations récurrentes

Résoudre les équations récurrentes suivantes. Pour (1), on pourra par exemple remarquer que la somme peut s'écrire comme un certain produit de convolution discret.

$$\forall n \in \mathbb{N},\ nu_n = \sum_{k=0}^{n} u_k \tag{1}$$

$$\forall n \in \mathbb{N},\ u_{n+3} = 2u_{n+2} + 4u_{n+1} - 8u_n + (-2)^n \tag{2}$$

(5 *points*).

4 Le théorème chinois

Soit $m_1, \ldots, m_k$ des entiers naturels non nuls deux à deux premiers entre eux (cela signifie que $PGCD(m_i, m_j) = 1$ si $i \neq j$). Soit $a_1, \ldots, a_k$ des entiers donnés. On s'intéresse au système de congruences suivant :

$$\begin{cases} x \equiv a_1 \mod m_1 \\ x \equiv a_2 \mod m_2 \\ \vdots \\ x \equiv a_k \mod m_k \end{cases} \tag{3}$$

Soit m le produit des m_i :

$$m = m_1 m_2 \ldots m_k$$

Le théorème chinois affirme que le système (3) possède une unique solution x dans l'ensemble

$$\mathbb{Z}_m = \{0, 1 \ldots, m-1\}$$

Nous allons prouver ce résultat, et même donner une formule explicite pour la solution x. Vous êtes guidé pas à pas.

1. Soit $m_i' = \frac{m}{m_i}$ et soit m_i'' l'inverse de m_i' modulo m_i (on choisit le représentant appartenant à $\mathbb{Z}_{m_i}$). Justifier l'existence de m_i'' et rappeler comment il se calcule en pratique (1 *point*).
2. On définit

$$x = \left(\sum_{i=1}^{k} a_i m_i' m_i'' \right) \mod m \tag{4}$$

   (a) Justifier que, pour $j \in \{1, \ldots, k\}$, $a_j m_j' m_j'' \equiv a_j \mod m_j$ (0,5 *point*).

   (b) Justifier que pour $i \neq j$, $a_i m_i' m_i'' \equiv 0 \mod m_j$ (0,5 *point*).

   (c) En déduire que x donné par (4) est une solution de (3) dans $\mathbb{Z}_m$ (1 *point*).
3. Soit f l'application de $\mathbb{Z}_m$ dans $\mathbb{Z}_{m_1} \times \ldots \times \mathbb{Z}_{m_k}$ qui à $x \in \mathbb{Z}_m$ associe

$$f(x) = (x \mod m_1, \ldots, x \mod m_k)$$

   Justifier que f est surjective, puis qu'elle est bijective. En déduire que la solution de (3) dans $\mathbb{Z}_m$ est unique (2 *points*).