Arithmétique et complexité

Durée : 2h.

23 Avril 2012

Exercice 1 : **8 points**

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 entier relatif quelconque et n un entier naturel.

Le principe de cet algorithme est très simple : 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. Voici l'algorithme d'exponentiation rapide :

```
Fonction res = exponentiationrapide (a,n)
x := a ;
N := n ;
Res := 1A ;
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
Finfonction
```

   (a) Vérifier que l'algorithme fait bien ce qu'on attend de lui, c'est à dire qu'il se termine et qu'en sortie la variable `res` contient a^n (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 pourcalculer efficacement $a^n \mod p$ 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) ?

Exercice 2 : **5 points**

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.

1. $\forall n \in \mathbb{N}, \quad nu_n = \sum_{k=0}^{n} u_k$.
2. $\forall n \in \mathbb{N}, \quad u_{n+3} = 2u_{n+2} + 4u_{n+1} - 8u_n + (-2)^n$.

Exercice 3 : **7 points**

1. Écrire 77 en base 2.
2. Décomposer 209 en facteurs premiers.
3. Calculer le pgcd d de 77 et 180 et déterminer des entiers u et v tels que $77u + 180v = d$.
4. Soit a et n des entiers relatifs. À quelle condition existe-t-il un entier x tel que $ax \equiv 1[n]$?
5. Déterminer tous les entiers x tels que $77x \equiv 1[180]$.
6. Trouver $a \in \{0, 1, \ldots, 18\}$ tel que $27^{77} \equiv a[19]$.
7. Trouver $b \in \{0, 1, \ldots, 10\}$ tel que $27^{77} \equiv b[11]$.
8. Calculer le reste de la division euclidienne de 27^{77} par 209.
9. Juliette et Roméo ont lu dans la revue *Pour la Science* un article sur le principe de cryptographie RSA. Ils décident de le tester sur un exemple simple pour vérifier qu'ils ont compris. Pour cela, Juliette choisit la clef publique ($n = 209$, $c = 7$; Roméo choisit alors un entier, compris entre 0 et 208, puis le code avant de transmettre à Juliette le résultat : 27.
   Pouvez-vous aider Juliette à retrouver l'entier choisi par Roméo ? Justifiez soigneusement votre réponse ; en particulier, rappelez le principe du codage et du décodage et calculez la clef secrète qui permet le décodage.
   Vous pourrez bien sûr utiliser les questions précédentes.