

Deuxième session, 27 juin 2006

Durée 3 heures. Documents interdits, calculatrices autorisées.

Exercice 1 – 17 est-il un carré modulo $209 = 11 \times 19$?

Exercice 2 – Trouver les solutions dans \mathbb{Z}^2 du système d'équations

$$\begin{cases} 2x + 5y \equiv 1 \pmod{8}, \\ 3x + 2y \equiv 0 \pmod{12}. \end{cases}$$

Exercice 3 – La lettre N désigne un entier > 1 . On dit que N vérifie la condition (C) de Carmichael si

$$(C) \quad a^N \equiv a \pmod{N} \quad \text{pour tout } a \in \mathbb{Z}.$$

1) Montrer que si N vérifie (C), alors il n'a pas de facteur carré autre que 1, et que $(p-1) \mid (N-1)$ pour tout p premier divisant N .

2) Réciproquement, montrer que tout N vérifiant les conditions du 1) vérifie (C).

3) Montrer que, si N vérifie (C) et n'est pas premier, alors il a au moins 3 diviseurs premiers.

4) On suppose que N vérifie (C), et qu'il est impair. On applique le test de non-primauté de Rabin-Miller à N et on suppose qu'il est positif, i.e. qu'on dispose de $a \in \mathbb{Z}/N\mathbb{Z}$ qui est témoin de non-primauté. Montrer qu'on peut facilement en déduire un facteur non trivial de N . [Combien de racines carrées de 1 connaît-on ?]

Problème (Racines modulo p)

Soit p un nombre premier *impair*. On exprimera les estimations de complexité en terme d'opérations élémentaires dans \mathbb{F}_p , en utilisant la notation \tilde{O} pour ne pas avoir à tenir compte des facteurs logarithmiques ou des constantes. On rappelle qu'une opération (+, \times , division euclidienne ou pgcd) sur deux polynômes de degré $\leq n$ dans $\mathbb{F}_p[X]$ utilise $\tilde{O}(n)$ opérations dans \mathbb{F}_p .

Soit $T \in \mathbb{F}_p[X]$ unitaire de degré $n > 0$, dont on désire trouver les racines dans \mathbb{F}_p .

1) Écrire une procédure MAPLE qui détermine les racines de T dans \mathbb{F}_p en calculant $T(0), T(1), \dots$. Quelle est sa complexité ?

2) Soit $D := \text{pgcd}(T, X^p - X)$.

a) Montrer que D est produit de facteurs linéaires dans $\mathbb{F}_p[X]$, que ses racines sont simples, et qu'il a les mêmes racines que T .

b) Montrer que D se calcule en $\tilde{O}(n \log p)$ opérations dans \mathbb{F}_p . [Travailler dans $\mathbb{F}_p[X]/(T)$.]

On suppose dorénavant que $D = T$ avec les notations de la question précédente, c'est-à-dire que T est scindé à racines simples, de degré $n > 1$.

3) Soit A la \mathbb{F}_p -algèbre $\mathbb{F}_p[X]/(T)$. On pose $t := (p-1)/2$.

a) Utiliser le lemme chinois pour montrer que $A \cong (\mathbb{F}_p)^n$ comme \mathbb{F}_p -algèbre.

b) Montrer que tout $a \in A$ vérifie $a^p = a$, et donc $a(a^t - 1)(a^t + 1) = 0$.

4) Si $a, a^t - 1, a^t + 1$ sont simultanément non-nuls, $a \in A$ est dit *bon*, et *mauvais* sinon.

a) Soit $u \in \mathbb{F}_p[X]$ tel que la classe de u dans A soit un diviseur de zéro¹. Montrer que $\text{pgcd}(u, T)$ est un facteur *propre* de T , c'est-à-dire différent de 1 et de T .

b) Montrer qu'un bon a permet de calculer un diviseur de 0 dans A , puis d'en déduire un facteur propre de T .

c) Un représentant de degré minimal dans $\mathbb{F}_p[X]$ de $a \in A$ étant donné, quelle est la complexité du test « a est-il bon ? »

d) Montrer que $a^t = 1$ et $a^t = -1$ ont tous deux t^n solutions $a \in A$.

e) En déduire que, si on choisit a uniformément au hasard dans A , la probabilité que a soit mauvais est $(2t^n + 1)/p^n$. Montrer que cette dernière quantité est toujours inférieure à $1/2$ ($p > 2, n > 1$).

5) Montrer que l'algorithme suivant est correct :

Entrées: $T \in \mathbb{F}_p[X]$ unitaire, scindé sur \mathbb{F}_p , p impair.

Sorties: la liste des racines de T .

1: Si $\deg T = 1$, retourner $-T(0)$.

2: **répéter**

3: tirer a uniformément au hasard dans $A = \mathbb{F}_p[X]/(T)$

4: **tant que** a est mauvais.

5: Soit D le facteur propre (unitaire) de T associé au bon a . Effectuer deux appels récursifs avec pour arguments D et T/D et retourner la concaténation des résultats.

6) On modélise le fonctionnement de l'algorithme pour une suite de tirages de a (supposés uniformes et indépendants) par un arbre dont chaque noeud est étiqueté par un polynôme T , et correspond au tirage d'un a . La racine est le polynôme d'origine. Si a est mauvais, le noeud a un fils unique étiqueté par le même T . Et deux noeuds étiquetés par D et T/D sinon.

a) Soit α_i et α_j deux racines distinctes de T . Montrer que, si on tire a uniformément au hasard dans A , il est bon et sépare α_i et α_j avec probabilité $1 - (1 + 2t^2)/p^2 \geq 1/2$.

b) En déduire que la probabilité que α_i et α_j ne soient pas séparées après k tirages est $\leq 2^{-k}$.

★ c) L'algorithme s'arrête une fois que les n racines sont séparées. Montrer que la probabilité p_k que l'on ne se soit pas arrêté après k tirages est $O(n^2 2^{-k})$, puis que l'espérance de la hauteur de l'arbre est $O(\log n)$. [Noter qu'il existe des suites infinies de tirages tels que l'algorithme ne termine pas. On vient de montrer qu'elles sont rares.]

★ d) Montrer que l'algorithme complet utilise en moyenne $\tilde{O}(n \log p)$ opérations pour factoriser T .

¹Si R est un anneau, on dit que $a \in R \setminus \{0\}$ est *diviseur de 0* s'il existe $b \in R \setminus \{0\}$ tel que $ab = 0$.