

**Examen, 29 avril 2009**

**Durée 3 heures. Documents interdits, calculatrices autorisées.**

Dans les estimations de complexité, on utilisera 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  $K[X]$  utilise  $\tilde{O}(n)$  opérations élémentaires dans le corps  $K$ . Une opération sur deux entiers de valeur absolue  $\leq 2^n$  utilise  $\tilde{O}(n)$  opérations élémentaires sur des chiffres. Dans les deux cas, on dira « en temps  $\tilde{O}(n)$  » au lieu de « en utilisant  $\tilde{O}(n)$  opérations élémentaires ».

**Exercice 1** – Soient  $p$  un nombre premier et  $a$  un entier vérifiant  $0 < a < p$ .

- 1) Montrer comment calculer l'inverse de  $a$  modulo  $p$  en utilisant l'identité  $a^{p-1} \equiv 1 \pmod{p}$ .
- 2) Rédiger l'algorithme correspondant. On fera bien sûr appel à l'exponentiation binaire.
- 3) Estimer, en fonction de  $p$ , la complexité algébrique (nombre d'opérations dans  $\mathbb{F}_p$ ), puis la complexité binaire de cet algorithme.
- 4) Rappeler comment se servir de l'algorithme d'Euclide étendu pour résoudre le même problème d'inversion, puis comparer les complexités binaires des deux procédés.

**Exercice 2** – Soit  $N > 1$  un entier dont on désire montrer qu'il est *premier*.

1) Soit  $p \mid N - 1$  un nombre premier et  $e = v_p(N - 1)$  la plus grande puissance de  $p$  divisant  $N - 1$ . On suppose qu'il existe  $a \in \mathbb{Z}$  vérifiant

- $a^{N-1} \equiv 1 \pmod{N}$ ,
- $\text{pgcd}(a^{(N-1)/p} - 1, N) = 1$ .

On va montrer que tout diviseur  $d$  de  $N$  vérifie  $d \equiv 1 \pmod{p^e}$ .

- a) Montrer qu'il suffit de démontrer l'assertion pour tout diviseur  $d$  *premier*.
- b) On suppose donc  $d \mid N$  premier et on note  $o$  l'ordre de  $a$  dans  $(\mathbb{Z}/d\mathbb{Z})^*$ . Montrer que  $o \mid N - 1$ , mais que  $o \nmid (N - 1)/p$ .
- c) En déduire que  $p^e \mid o$  et conclure.

2) On suppose que  $N - 1 = FU$ , où  $F \geq \sqrt{N}$  est un facteur dont tous les diviseurs premiers sont connus tel que  $(F, U) = 1$ , et que pour chaque  $p \mid F$ , on connaît  $a(p)$  vérifiant les propriétés du 1). Soit  $d > 1$  un diviseur de  $N$ .

- a) Montrer que  $d \equiv 1 \pmod{F}$ . [Utiliser le 1) et penser au Lemme Chinois.]
- b) En déduire que  $d > \sqrt{N}$ , puis que  $N$  est premier.

3) On suppose  $N$  premier, et on fixe  $p \mid N - 1$ . Tirant  $a$  uniformément au hasard dans  $[1, N]$ , quelle est la probabilité de trouver un  $a(p)$  vérifiant la propriété ci-dessus ?

### Problème

On note «  $a \bmod b$  » le reste de la division euclidienne de  $a$  par  $b$ .

- 1) Soit  $P \in K[X]$  non constant, et  $\mathcal{L} = [x_0, \dots, x_{n-1}] \in K^n$  une liste d'éléments de  $K$ .
  - a) Écrire une procédure MAPLE qui teste si  $P(x_i) \neq 0$  pour tout  $0 \leq i < n$ .
  - b) Majorer la complexité de cette procédure, en fonction de  $\deg P$  et  $n$ .
- 2) Soit  $N > 0$  un entier et  $\mathcal{L} = [x_0, \dots, x_{n-1}] \in \mathbb{Z}^n$  une liste de  $n$  entiers  $> 1$ .
  - a) Écrire une procédure MAPLE qui teste si  $N$  n'est divisible par aucun  $x \in \mathcal{L}$ .
  - b) Expliquer pourquoi on peut supposer que les éléments de  $\mathcal{L}$  sont inférieurs à  $N$ . Sous cette hypothèse, majorer la complexité de cette procédure, en fonction de  $\log N$  et  $n$ .
- 3) Expliquer en quoi les deux questions précédentes résolvent essentiellement le même problème.
- 4) On se concentre désormais sur le cas  $N \in \mathbb{Z}$ ,  $\mathcal{L} \in \mathbb{Z}^n$ , plus facile à décrire, et on s'intéresse plus généralement à l'ensemble des  $N \bmod \mathcal{L}[i]$ . On suppose dans toute la suite que  $\#\mathcal{L} = n = 2^k$  est une puissance de 2.
  - a) Montrer que l'hypothèse  $n = 2^k$  est inoffensive : on peut toujours se ramener à cette situation.

- b) Pour  $0 \leq i \leq k$  et  $0 \leq j < 2^{k-i}$ , on pose  $M_{i,j} = \prod_{0 \leq \ell < 2^i} \mathcal{L}[j2^i + \ell]$ .

Si  $k = 3$ , dessiner l'arbre binaire naturel dont les noeuds sont les  $M_{i,j}$  et tel que chaque noeud contienne le produit de ses deux fils.

- c) Montrer que l'ensemble des  $M_{i,j}$  se calculent en temps  $\tilde{O}(\log \mathcal{L})$ , où la taille totale «  $\log \mathcal{L}$  » de la liste  $\mathcal{L}$  est définie par  $\log \mathcal{L} := \sum_{i \leq n} \log \mathcal{L}[i]$ .

- d) Écrire une procédure MAPLE calculant tous les  $M_{i,j}$ .

5) On suppose que les  $M_{i,j}$  sont précalculés, stockés sur un arbre<sup>1</sup> organisé de telle sorte que l'on puisse détacher le sous-arbre gauche ou droit de la racine, respectivement associés à la première ou deuxième moitié de  $\mathcal{L}$ , en temps négligeable. On considère l'algorithme suivant

---

**Entrées:** Un arbre des  $M_{i,j}$  associé à une liste  $\mathcal{L} \in \mathbb{Z}^n$ ,  $n = 2^k$  ; un entier  $N > 0$ ,  $\log N < \log \mathcal{L}$ .

**Sorties:** La liste des  $N \bmod \mathcal{L}[i]$ ,  $0 \leq i < n$ .

- 1: Si  $n = 1$ , retourner  $N \bmod \mathcal{L}[0]$ .
- 2: Soit  $r_0 \leftarrow N \bmod M_{k-1,0}$ . Calculer récursivement les  $r_0 \bmod \mathcal{L}[i]$ ,  $0 \leq i < n/2$ .
- 3: Soit  $r_1 \leftarrow N \bmod M_{k-1,1}$ . Calculer récursivement les  $r_1 \bmod \mathcal{L}[i]$ ,  $n/2 \leq i < n$ .
- 4: Renvoyer la concaténation des résultats.

---

a) Détailler le passage « Calculer récursivement... ». Avec quelles entrées rappelle-t-on la fonction ?

- b) Montrer que l'algorithme est correct et calcule les  $N \bmod \mathcal{L}[i]$  en temps  $\tilde{O}(\log \mathcal{L})$ .

6) On *admet* les deux faits suivants :

- Le crible d'Ératosthène calcule  $\mathcal{L} := \{p \leq x : p \text{ premier}\}$  en temps  $\tilde{O}(x)$ ,
- On a  $\log \mathcal{L} = \sum_{p \leq x} \log p \sim x$  quand  $x \rightarrow \infty$  (théorème des nombres premiers).

- a) Comment trouver tous les facteurs premiers de  $N$  inférieurs à  $\log N$  en temps  $\tilde{O}(\log N)$  ?
- b) Comparer avec la méthode naïve de division successive par les éléments de  $\mathcal{L}$ .
- c) Utilisant la méthode récursive, à quel coût détecte-t-on *tous* les facteurs premiers de  $N$  (dans le cas le pire, que l'on explicitera) ?

---

<sup>1</sup>On n'explicitera pas l'implantation de cet arbre ; la représentation standard d'un arbre binaire parfait par un tableau unidimensionnel convient.