18 juin 2024
Master 1 M.F., Orsay

Examen de M.A.O. Calcul Formel Durée : 3 heures

Lorsqu'on évoque le coût d'un algorithme, on attend toujours une réponse de la forme $O(\cdots)$: déterminer la constante implicite dans le symbole $O(\cdots)$ n'est pas demandé. De plus, sauf mention explicite du contraire, on compte le nombre d'opérations arithmétiques (additions, multiplications, soustractions, divisions), dans $\mathbb{Z}$ ou dans un corps selon le contexte.

Il est autorisé d'admettre le résultat de certaines questions pour traiter les suivantes. L'examen est constitué d'un unique problème. La partie 1 décrit les notations, le contexte, et contient des questions préliminaires. Les parties 2, 3 et 4 (qui sont indépendantes entre elles, sauf la question 22) étudient différents aspects.

Partie 1 : notations et préliminaires

Dans tout ce problème on fixe un entier impair $n \geq 3$. On note r l'ordre de 2 dans le groupe multiplicatif $(\mathbb{Z}/n\mathbb{Z})^\times$, c'est-à-dire le plus petit entier strictement positif r tel que n divise $2^r - 1$. On fixe un corps fini à 2^r éléments, noté $\mathbb{F}_{2^r}$, et une racine primitive n-ième de l'unité dans $\mathbb{F}_{2^r}$, notée α.

1. Justifier l'existence d'une racine primitive n-ième de l'unité dans $\mathbb{F}_{2^r}$.
2. Rappeler brièvement comment on peut représenter informatiquement un élément de $\mathbb{F}_{2^r}$, et calculer le produit de deux tels éléments. En fonction de r, quel est le coût de ce calcul en opérations élémentaires sur les bits ?
3. Dans cette question seulement, on suppose que n est un nombre premier. Étant donné $\alpha \in \mathbb{F}_{2^r}$, comment peut-on tester efficacement si c'est une racine primitive n-ième de l'unité ? Quel est le coût de ce calcul, en nombre d'opérations arithmétiques dans $\mathbb{F}_{2^r}$?

On fixe un entier $t \geq 1$ et on note

$$\Sigma = \left\{ \overline{2^i k},\ i \in \mathbb{N},\ 1 \leq k \leq 2t \right\} \subset \mathbb{Z}/n\mathbb{Z}$$

où $\overline{2^i k}$ désigne la classe de $2^i k$ modulo n. On considère le polynôme

$$g(X) = \prod_{\sigma \in \Sigma} (X - \alpha^\sigma)$$

dont on sait qu'il appartient à $\mathbb{F}_2[X]$, et on pose

$$s = \deg g = \operatorname{Card} \Sigma.$$

4. Justifier que la notation α^σ, avec $\sigma \in \mathbb{Z}/n\mathbb{Z}$, a bien un sens.
5. On calcule $g(X)$ en partant du polynôme constant 1, et en multipliant successivement par $X - \alpha^\sigma$ pour σ parcourant Σ. On suppose que les α^σ sont déjà disponibles (par exemple qu'ils sont stockés dans une table). En fonction de s, quel est le coût du calcul de $g(X)$ en nombre d'opérations arithmétiques dans $\mathbb{F}_{2^r}$?

On considère le sous-espace vectoriel C de $\mathbb{F}_2[X]_{<n}$ défini par

$$C = \{gT,\, T \in \mathbb{F}_2[X],\, \deg T < n - s\}.$$

On identifie l'espace $\mathbb{F}_2[X]_{<n}$ des polynômes de degré au plus $n-1$ à coefficients dans $\mathbb{F}_2$ avec $\mathbb{F}_2^n$, via l'isomorphisme linéaire qui à $x_1X^{n-1} + \ldots + x_{n-1}X + x_n$ associe $(x_1, \ldots, x_n)$.

6. Combien valent la dimension et la longueur du code C ?
7. Déduire de la question précédente une majoration de la distance minimale de C.

Partie 2 : minoration de la distance minimale de C.

Dans cette partie on va minorer la distance minimale du code C défini dans la partie 1.

8. Soient $\varrho_1, \ldots, \varrho_w$ des éléments deux à deux distincts de $\mathbb{F}_{2^r} \setminus \{0\}$, avec $w \geq 1$. Démontrer que le déterminant de la matrice $R = [\varrho_j^i]_{1 \leq i,j \leq w}$ est non nul.
9. Soient $a_1, \ldots, a_w$ des entiers naturels deux à deux distincts, compris au sens large entre 0 et $n-1$. En utilisant la question précédente, démontrer que le polynôme $X^{a_1} + \cdots + X^{a_w}$ ne peut pas s'annuler en α^i pour tout $i \in \{1, \ldots, w\}$.
10. Démontrer que la distance minimale du code C défini dans la partie 1 est supérieure ou égale à $2t + 1$.
11. Combien d'erreurs le code C peut-il détecter ? Combien peut-il en corriger ?

Partie 3 : un exemple.

On reprend les notations de la partie 1 dans le cas particulier $n = 15$, $r = 4$, $t = 1$. On implémentera $\mathbb{F}_{2^4}$ comme étant $\mathbb{F}_2[X]/(X^4 + X + 1)$ et on note α la classe de X dans ce quotient.

12. ($\star$) Vérifier que $\mathbb{F}_2[X]/(X^4 + X + 1)$ est un corps. *Dans cette question comme dans les suivantes, on pourra utiliser n'importe quelle commande de Sage.*
13. ($\star$) Vérifier que α est une racine primitive n-ième de l'unité.
14. ($\star$) Calculer Σ, s et $g(X)$; vérifier que $g(X) = X^4 + X + 1$.
15. Déterminer une matrice génératrice de C.
16. ($\star$) Calculer la distance minimale de C. *On pourra consulter l'aide en ligne de commande de la fonction* `codes.CyclicCode` *et l'utiliser librement pour représenter le code C. On pourra également utiliser les commandes* `C.generator_matrix()` *pour obtenir la matrice génératrice du code C ainsi que* `C.minimum_distance()` *pour obtenir sa distance minimum.*

Partie 4 : un algorithme de décodage.

On reprend les notations de la partie 1. Soit $m \in \mathbb{F}_2[X]$ tel que $\deg m < n - s$ (où s a été défini dans la partie 1). Soit $e \in \mathbb{F}_2[X]_{<n}$ de poids inférieur ou égal à t ; on rappelle qu'on identifie $\mathbb{F}_2[X]_{<n}$ à $\mathbb{F}_2^n$. On pose $\varrho = mg + e$.

17. Rappeler l'interprétation, dans le contexte des codes correcteurs, de m, mg, e, ϱ.

Notons w le poids de e, avec $w \leq t$. On peut écrire

$$e = X^{i_1} + \cdots + X^{i_w} \text{ avec } 0 \leq i_1 < \cdots < i_w \leq n-1.$$

On considère les polynômes suivants de $\mathbb{F}_{2^r}[Y]$:

$$S(Y) = \sum_{k=1}^{2t} \varrho(\alpha^k) Y^{k-1}, \qquad \sigma(Y) = \prod_{j=1}^{w} (1 - \alpha^{i_j} Y), \qquad \tau(Y) = \sum_{j=1}^{w} \alpha^{i_j} \prod_{\substack{1 \leq \ell \leq w \\ \ell \neq j}} (1 - \alpha^{i_\ell} Y).$$

18. En justifiant que $\varrho(\alpha^k) = e(\alpha^k)$ pour tout $k \in \{1, \ldots, 2t\}$, démontrer que

$$S(Y) = \sum_{j=1}^{w} \sum_{k=1}^{2t} \alpha^{k i_j} Y^{k-1}$$

et en déduire que

$$S(Y)\sigma(Y) \equiv \tau(Y) \bmod Y^{2t}.$$

On considère maintenant l'algorithme d'Euclide étendu appliqué dans $\mathbb{F}_{2^r}[Y]$ à $r_0 = Y^{2t}$ et $r_1 = S(Y)$. Tant que $r_i \neq 0$ on note q_{i+1} le quotient, et r_{i+1} le reste, dans la division euclidienne de r_{i-1} par r_i. On pose $u_{i+1} = u_{i-1} - q_{i+1}u_i$ et $v_{i+1} = v_{i-1} - q_{i+1}v_i$ avec $u_0 = v_1 = 1$ et $u_1 = v_0 = 0$. En notant p le plus petit entier tel que $r_{p+1} = 0$, on a $r_p = \operatorname{pgcd}(r_0, r_1)$ et on rappelle les propriétés suivantes, valables pour tout $i \in \{0, \ldots, p-1\}$:

$$r_i = u_i r_0 + v_i r_1, \qquad \deg v_i = \deg r_0 - \deg r_{i-1}, \qquad \operatorname{pgcd}(u_i, v_i) = 1.$$

On note j l'unique entier vérifiant $\deg r_j < t \leq \deg r_{j-1}$.

19. Démontrer que le polynôme $r_j\sigma - v_j\tau$ est divisible par un certain polynôme de degré plus grand que le sien, et en déduire que c'est le polynôme nul.
20. En utilisant plusieurs fois la question précédente et en travaillant éventuellement modulo $v_j(Y)Y^{2t}$, montrer que

$$\sigma(Y) = \frac{v_j(Y)}{v_j(0)} \qquad \text{et} \qquad \tau(Y) = \frac{r_j(Y)}{v_j(0)}.$$

21. Donner un algorithme de décodage fondé sur le résultat de la question 20.
22. ($\star$) Implémenter en Sage une fonction `decodage` qui prend en argument un message $m \in \mathbb{F}_2^n$ et implémente l'algorithme de la question précédente dans le cas du code de la partie 3. Décoder le message 110011100110111.