ÉCOLES NORMALES SUPÉRIEURES

CONCOURS D'ADMISSION 2011 FILIÈRE MP – CONCOURS INFO

COMPOSITION D'INFORMATIQUE-MATHÉMATIQUES – (ULC)

(Durée : 4 heures)

Les calculatrices sont interdites.

La qualité de la rédaction sera prise en compte dans la note finale.

⋆ ⋆ ⋆

Autour des codes correcteurs

Nous allons traiter dans ce sujet quelques points relatifs aux codes correcteurs d'erreur. Ce domaine s'occupe de la transmission fiable d'informations via un canal possiblement bruité. Cette théorie trouve de nombreuses applications pratiques comme la transmission d'information par satellite ou les disques compacts par exemple.

Nous allons dans un premier temps examiner certains points fondamentaux de cette théorie.

Dans une deuxième partie, nous allons considérer une famille particulière de codes. Ils sont normalement définis sur des objets que l'on appelle corps finis mais pour des raisons de compatibilité avec le programme de l'épreuve, nous allons les définir sur $\mathbb{R}$ ou $\mathbb{C}$. Cela ne nous empêchera pas de mettre en évidence un certain nombre de leurs propriétés classiques. Nous allons notamment étudier un algorithme de décodage de ces codes.

La troisième partie sera consacrée au développement d'algorithmes qui nous permettront d'accélérer la procédure de décodage introduite dans la deuxième partie.

Pour finir, nous reviendrons dans le cadre plus réaliste de codes définis sur un ensemble fini et nous mettrons en place les propriétés qui nous permettront d'établir des bornes sur le nombre de mots que peut contenir un code, et d'avoir ainsi une idée un peu plus précise de l'efficacité de ce code.

Préambule

Dans la suite, si x est un nombre réel, $\lfloor x \rfloor$ désignera la partie entière de x, i.e. le plus grand entier relatif inférieur ou égal à x. Si E désigne un ensemble fini, on notera $\mathrm{card}(E)$ son nombre d'éléments. Soit m et n deux entiers relatifs, on définit

$$\binom{n}{m} = \begin{cases} \dfrac{n!}{(n-m)!m!} \text{ si } 0 \leqslant m \leqslant n, \\ 0 \text{ sinon.} \end{cases}$$

Une somme et un produit indexés sur un ensemble vide vaudront, comme d'habitude, respectivement 0 et 1.

Partie 1 : Codes correcteurs

Soit $\mathcal{A}$ un ensemble non vide, n un entier naturel non nul. On désignera par $\mathcal{A}^n$ l'ensemble $\{x = (x_1, x_2, \ldots, x_n), x_i \in \mathcal{A}\}$. Les éléments de $\mathcal{A}^n$ seront appelés mots.

Si $x = (x_1, x_2, \ldots, x_n)$ et $y = (y_1, y_2, \ldots, y_n)$ désignent deux mots de $\mathcal{A}^n$, on note

$$\delta_{H,n}(x, y) = \mathrm{card}(\{i \in \{1, 2, \ldots, n\}, x_i \neq y_i\}).$$

Question 1.1. Démontrer que $\delta_{H,n}$ définit une distance sur $\mathcal{A}^n$.

Dans la suite, $\delta_{H,n}$ sera notée plus simplement δ_H.

Un code sur $\mathcal{A}$ de longueur n est un sous-ensemble C de $\mathcal{A}^n$. Dans tout le problème, un code comportera toujours au moins deux éléments distincts. L'ensemble $\mathcal{A}$ sera appelé alphabet, l'entier n sera appelé la longueur du code C et les éléments de C seront appelés les mots du code.

Soit e un entier naturel. On dit qu'un code C de longueur n sur un alphabet $\mathcal{A}$ vérifie la condition de décodage d'ordre e si pour tout $y \in \mathcal{A}^n$, il existe au plus un mot $x \in C$ tel que $\delta_H(x, y) \leqslant e$.

La distance minimale d'un code C est l'entier

$$d_C = \min\{\delta_H(x, y); (x, y) \in C \times C, x \neq y\}.$$

Question 1.2. Justifier que d_C est bien définie.

Question 1.3. Soit C un code et e un entier naturel. Montrer que si $d_C \geqslant 2e+1$ alors C vérifie la condition de décodage d'ordre e.

La capacité de correction d'un code C est l'entier $e = \left\lfloor \frac{d_C - 1}{2} \right\rfloor$, où d_C désigne la distance minimale de C.

Dans la pratique, un mot x d'un code C de longueur n est envoyé via un canal de communication et l'on note y le mot de $\mathcal{A}^n$ reçu. Le but est, lorsque c'est possible, de retrouver x à partir de y : tout moyen permettant d'effectuer une telle opération est appelé décodage.

On suppose $\mathcal{A}$ fini pour toute la suite de la Partie 1.

Question 1.4. Soit $x \in \mathcal{A}^n$, $r \in \mathbb{N}$, on désigne par $B(x, r)$ la boule fermée de centre x et de rayon r pour la distance δ_H, i.e. $B(x, r) = \{y \in \mathcal{A}^n; \delta_H(x, y) \leqslant r\}$.

(a). Pour tout $i \in \mathbb{N}$, on désigne par $S(x, i)$ la sphère centrée en x et de rayon i, i.e. $S(x, i) = \{y \in \mathcal{A}^n; \delta_H(x, y) = i\}$. Démontrer que

$$\mathrm{card}(B(x, r)) = \sum_{i=0}^{r} \mathrm{card}(S(x, i)).$$

(b). En déduire que

$$\mathrm{card}(B(x, r)) = \sum_{i=0}^{r} (\mathrm{card}(\mathcal{A}) - 1)^i \binom{n}{i}.$$

Question 1.5. Soit C un code de longueur n sur un alphabet $\mathcal{A}$. Démontrer qu'il existe un plus petit rayon r tel que l'ensemble des boules de rayon r centrées en chaque mot du code forme un recouvrement de $\mathcal{A}^n$. On note $\rho(C)$ ce rayon.

Question 1.6.

(a). Soit C un code de longueur n sur un alphabet $\mathcal{A}$, de capacité de correction e. Démontrer que

$$\mathrm{card}(C) \sum_{i=0}^{e} (\mathrm{card}(\mathcal{A}) - 1)^i \binom{n}{i} \leqslant (\mathrm{card}(\mathcal{A}))^n.$$

(b). Soit C un code de longueur n sur un alphabet $\mathcal{A}$. Montrer de même que

$$\mathrm{card}(C) \sum_{i=0}^{\rho(C)} (\mathrm{card}(\mathcal{A}) - 1)^i \binom{n}{i} \geqslant (\mathrm{card}(\mathcal{A}))^n.$$

Intermède : notations et définitions

Dans les deux prochaines parties, $\mathbf{K}$ désignera le corps des nombres réels $\mathbb{R}$ ou le corps des nombres complexes $\mathbb{C}$, $\mathbf{K}[X]$ désignera l'anneau des polynômes à coefficients dans $\mathbf{K}$ et $\mathbf{K}(X)$

le corps des fractions rationnelles à coefficients dans $\mathbf{K}$. Ces deux ensembles sont aussi des $\mathbf{K}$-espaces vectoriels. Un élément P d'un de ces deux ensembles pourra aussi être noté $P(X)$. Un polynôme à coefficients dans $\mathbf{K}$ est dit unitaire si son coefficient dominant, i.e. le coefficient de son terme de plus haut degré, vaut 1. On convient que $\deg 0 = -\infty$.

Soit A, B et $C \in \mathbf{K}[X]$. On dit que A divise B s'il existe $D \in \mathbf{K}[X]$ tel que $B = AD$. On notera $A \equiv B \mod C$ si C divise $A - B$.

Il vous sera demandé d'estimer l'efficacité de certaines procédures de calculs. On le fera comme suit. Soit $\mathbf{K}$ désignant $\mathbb{R}$ ou $\mathbb{C}$, on supposera que le coût d'une opération arithmétique (addition, soustraction, multiplication, division) sur $\mathbf{K}$ est unitaire. On devra donc estimer, ou au moins majorer, le nombre d'opérations arithmétiques sur $\mathbf{K}$ qui seront effectuées au cours de l'exécution des algorithmes.

Soit $(f(n))$ et $(g(n))$ deux suites réelles définies à partir d'un certain rang n_0. On écrit que

$$f(n) = O(g(n)),\ n \to \infty,$$

s'il existe $K > 0$ et $N \geqslant n_0$ tels que pour tout $n \in \mathbb{N}$, $n \geqslant N$, on a $|f(n)| \leqslant K|g(n)|$.

On généralise cette notation à des fonctions de deux arguments : soit m_0 et n_0 deux entiers fixés, si f et g sont deux fonctions à valeurs réelles définies en tout couple $(m, n) \in \mathbb{N}^2$, $m \geqslant m_0$, $n \geqslant n_0$. On écrit que

$$f(m,n) = O(g(m,n)),\ n \to \infty,\ m \to \infty,$$

s'il existe $K > 0$, $N \geqslant \max(m_0, n_0)$ tels que pour tout n et $m \in \mathbb{N}$, $\min(n, m) \geqslant N$, on a $|f(m,n)| \leqslant K|g(m,n)|$.

À titre d'exemple, on va effectuer une majoration du coût du produit de deux polynômes. On va au préalable admettre les résultats suivants : soit $A = \sum_{0 \leqslant i \leqslant n} a_i X^i$ et $B = \sum_{0 \leqslant j \leqslant m} b_j X^j \in \mathbf{K}[X]$ avec $a_n b_m \neq 0$, $\lambda \in \mathbf{K}$,

- la multiplication de A par un monôme X^k, $k \in \mathbb{N}$, n'a pas de coût arithmétique (le polynôme obtenu est $\sum_{0 \leqslant i \leqslant n} a_i X^{i+k}$) ;
- la multiplication de A par λ coûte au plus $\deg A + 1$ multiplications dans $\mathbf{K}$ (le polynôme obtenu est $\sum_{0 \leqslant i \leqslant n} (\lambda a_i) X^i$) ;
- l'addition de A et B coûte au plus $\min(\deg A, \deg B) + 1$ additions dans $\mathbf{K}$ (supposons $n \geqslant m$, on a $A + B = \sum_{0 \leqslant k \leqslant n} c_k X^k$ avec $c_k = a_k + b_k$ pour $k \leqslant m$ et $c_k = a_k$ sinon).

Soit deux polynômes, A et $B \in \mathbf{K}[X]$. On peut calculer le produit, noté C, de A et B de la manière élémentaire suivante :

Algorithme 1

Entrée : n et $m \in \mathbb{N}$, $A = \sum_{0 \leqslant i \leqslant n} a_i X^i$ et $B = \sum_{0 \leqslant j \leqslant m} b_j X^j \in \mathbf{K}[X]$.
Sortie : Un polynôme $C \in \mathbf{K}[X]$ égal au produit de A et B.

1. $C_0 \longleftarrow \sum_{0 \leqslant j \leqslant m} a_0 b_j X^j$
2. Pour $i = 1, \ldots, n$, faire
3. $\quad S_i \longleftarrow (a_i X^i) B$
   $\quad C_i \longleftarrow C_{i-1} + S_i$
4. Renvoyer $C = C_n$

Le symbole $\longleftarrow$ dans un algorithme signifie «prend la valeur de». L'expression «renvoyer XXX» dans un algorithme signifie «sortir de l'algorithme et retourner XXX».

L'étape 1 coûte au plus $m+1$ produits (les $a_0 b_j$ pour $j = 0, \ldots, m$) dans $\mathbf{K}$.

Soit $i \in \{1, \ldots, n\}$, on constate que $S_i = \sum_{i \leqslant j \leqslant i+m} a_i b_{j-i} X^j$ et on montre par récurrence que $\deg C_{i-1} \leqslant m + i - 1$. On écrit donc $C_{i-1} = \sum_{0 \leqslant j \leqslant m+i-1} c_{i-1,j} X^j$. Le polynôme $C_i = C_{i-1} + S_i$ est donc de la forme $\sum_{0 \leqslant j \leqslant m+i} c_{i,j} X^j$ avec

$$c_{i,j} = \begin{cases} c_{i-1,j} \text{ pour } j = 0, \ldots, i-1, \\ c_{i-1,j} + a_i b_{j-i} \text{ pour } j = i, \ldots, m+i-1, \\ a_i b_m \text{ pour } j = m+i. \end{cases}$$

Donc, pour $j = 0, \ldots, i-1$, il n'y a pas de coût arithmétique. Pour $j = i, \ldots, m+i-1$, on calcule au plus un produit et une addition dans $\mathbf{K}$ et pour $j = m+i$, on effectue au plus une multiplication. L'étape 3 a donc un coût total d'au plus $m+1$ multiplications et m additions dans $\mathbf{K}$.

L'étape 2 consistant à répéter n fois l'étape 3, son exécution nécessite au plus $n(m+1)$ multiplications et nm additions dans $\mathbf{K}$.

Il y a donc un coût total d'au plus $m + 1 + n(m+1) + nm = (m+1)(n+1) + nm$ opérations arithmétiques dans $\mathbf{K}$. Comme $(m+1)(n+1) + nm \leqslant (2m)(2n) + nm = 5mn$ dès que $\min(n, m) \geqslant 1$, on en déduit que le nombre total d'opérations arithmétiques sur $\mathbf{K}$ est en $O(mn)$.

Partie 2 : Décodage d'une famille de codes

Dans cette partie, l'alphabet sera le corps $\mathbf{K}$.

Pour $n \in \mathbb{N}$, $n \geqslant 1$, l'ensemble $\mathcal{A}^n$ est donc $\mathbf{K}^n$, l'espace vectoriel sur $\mathbf{K}$ muni des opérations standard $+$ et $\cdot$ définies par :
- pour $x = (x_1, \ldots, x_n)$ et $y = (y_1, \ldots, y_n)$ dans $\mathcal{A}^n$, on a $x + y = (x_1 + y_1, \ldots, x_n + y_n)$;
- pour $x = (x_1, \ldots, x_n)$ dans $\mathcal{A}^n$ et $\lambda \in \mathbf{K}$, on a $\lambda \cdot x = (\lambda x_1, \ldots, \lambda x_n)$.

Si $x \in \mathcal{A}^n$, on notera $w_H(x) = \delta_H(x, 0)$.

Soit $k \in \mathbb{N}, k \geqslant 1$, un code linéaire de longueur n sur $\mathbf{K}$ et de dimension k (on dira un code $[n, k]$) est un sous-espace vectoriel de $\mathbf{K}^n$ de dimension k.

Question 2.1. Soit un code linéaire C, w_H est-elle une norme sur le $\mathbf{K}$-espace vectoriel C ? Justifier votre réponse.

Question 2.2. Démontrer que la distance minimale d'un code linéaire C est égale au minimum de $w_H(x)$, où x décrit $C \setminus \{0\}$.

Question 2.3. *Borne de Singleton.* Démontrer que tout code $[n, k]$ de distance minimale d vérifie l'inégalité $k + d \leqslant n + 1$. On pourra faire intervenir le sous-ensemble des points de $\mathbf{K}^n$ dont les $n - d + 1$ premières composantes sont nulles.

Soit n un entier naturel non nul. Soit $\alpha_1, \ldots, \alpha_n \in \mathbf{K}$, deux à deux distincts. La fonction d'évaluation associée à $\alpha = (\alpha_1, \ldots, \alpha_n)$ est

$$\begin{array}{rcl} \mathrm{ev}_\alpha : \mathbf{K}[X] & \to & \mathbf{K}^n \\ f & \mapsto & (f(\alpha_1), \ldots, f(\alpha_n)). \end{array}$$

Soit $k \in \mathbb{N}$, $1 \leqslant k \leqslant n$, soit

$$L_k = \{f \in \mathbf{K}[X] \,;\, \deg f < k\}.$$

Soit $\mathrm{RS}_{\alpha,k}$ défini par

$$\mathrm{RS}_{\alpha,k} = \mathrm{ev}_\alpha(L_k) = \{\mathrm{ev}_\alpha(f); f \in L_k\}.$$

Question 2.4. Démontrer que $\mathrm{RS}_{\alpha,k}$ est un code $[n, k]$.

Question 2.5.

(a). Démontrer que, pour tout $f \in L_k \setminus \{0\}$, le vecteur $\mathrm{ev}_\alpha(f)$ a au plus $k - 1$ composantes nulles.

(b). Déterminer la distance minimale de $\mathrm{RS}_{\alpha,k}$. Que remarque-t-on ?

Nous allons maintenant étudier un procédé de décodage des codes $\mathrm{RS}_{\alpha,k}$. On se donne un entier e tel que $0 < e < (n - k + 1)/2$.

Algorithme 2

Entrée : Le mot reçu $y = (y_1, \ldots, y_n) \in \mathbf{K}^n$

Sortie : Un polynôme $f \in \mathbf{K}[X]$ tel que $\deg f < k$ et $\delta_H(\mathrm{ev}_\alpha(f), y) \leqslant e$ ou «échec»

1. Calculer $E, F \in \mathbf{K}[X]$ tels que
   a. $F(\alpha_i) - y_i E(\alpha_i) = 0$ pour tout $i \in \{1, \ldots, n\}$
   b. $\deg F < e + k$
   c. $\deg E = e$ et E est unitaire

   Si de tels polynômes n'existent pas, renvoyer «échec»
2. Si E ne divise pas F, renvoyer «échec». Sinon, calculer $f = F/E$.
3. Si $\delta_H(\mathrm{ev}_\alpha(f), y) > e$, renvoyer «échec». Sinon, renvoyer f.

Question 2.6. Soit $f \in \mathbf{K}[X]$ tel que $\deg f < k$ et $\delta_H(\mathrm{ev}_\alpha(f), y) \leqslant e$, exhiber un couple $(E, F) \in \mathbf{K}[X]^2$ vérifiant les conditions 1a, 1b et 1c de l'Algorithme 2, et tel que $f = F/E$.

Question 2.7. Soit (E_1, F_1) et (E_2, F_2) deux paires de polynômes satisfaisant les conditions 1a, 1b et 1c de l'Algorithme 2. Démontrer que

$$\frac{F_1(X)}{E_1(X)} = \frac{F_2(X)}{E_2(X)}.$$

Question 2.8. On suppose que l'on sait effectuer l'étape 1 de l'Algorithme 2. Démontrer que s'il existe $f \in \mathbf{K}[X]$ tel que $\deg f < k$ et $\delta_H(\mathrm{ev}_\alpha(f), y) \leqslant e$, l'Algorithme 2 le calcule et renvoie «échec» sinon.

Soit $n \in \mathbb{N}$. On dira qu'un système d'équations linéaires à coefficients dans $\mathbf{K}$ est de taille au plus n si son nombre d'inconnues et son nombre d'équations sont tous deux inférieurs ou égaux à n. On admet que la résolution d'un système d'équations linéaires à coefficients dans $\mathbf{K}$ de taille au plus n peut s'effectuer en un coût $O(n^3)$ d'opérations arithmétiques sur $\mathbf{K}$.

Question 2.9. Démontrer que le calcul effectif de polynômes E et F vérifiant les conditions 1a, 1b et 1c de l'Algorithme 2 peut se ramener à la résolution d'un système d'équations linéaires à coefficients dans $\mathbf{K}$, que l'on explicitera, de taille au plus n.

Si P est un polynôme, on notera $\mathrm{cd}(P)$ son coefficient dominant.

Algorithme 3

Entrée : $A = \sum_{0 \leqslant i \leqslant n} a_i X^i$ et $B = \sum_{0 \leqslant i \leqslant m} b_i X^i \in \mathbf{K}[X]$ avec $b_m \neq 0$
Sortie : Le couple $(Q, R) \in \mathbf{K}[X]$ tel que $A = BQ + R$ et $\deg R < m$

1. Si $n < m$, renvoyer $Q = 0$ et $R = A$
2. $R \longleftarrow A, u \longleftarrow b_m^{-1}$
3. pour $i = n - m, n - m - 1, \ldots, 0$, faire
4. si $\deg R = m + i$ alors $q_i \longleftarrow \mathrm{cd}(R)u$, $R \longleftarrow R - q_i X^i B$
   sinon $q_i \longleftarrow 0$
5. Renvoyer $Q = \sum_{0 \leqslant i \leqslant n-m} q_i X^i$ et R

Question 2.10. *Division euclidienne.* Soit $n, m \in \mathbb{N}$, A et $B \in \mathbf{K}[X]$ avec $\deg A = n$ et $\deg B = m$.

(a). Démontrer qu'un couple $(Q, R) \in \mathbf{K}[X]^2$ tel que $A = BQ + R$ et $\deg R < m$ est unique.

(b). Démontrer que l'Algorithme 3 retourne bien la sortie annoncée.

(c). Démontrer que, si $n \geqslant m$, cet algorithme nécessite au plus $(2m+1)(n-m+1)+1$ opérations arithmétiques dans $\mathbf{K}$.

Les polynômes Q et R renvoyés par l'Algorithme 3 seront appelés respectivement quotient et reste de la division euclidienne de A par B.

Question 2.11. Démontrer que l'Algorithme 2 peut s'exécuter en un nombre $O(n^3)$ d'opérations arithmétiques sur $\mathbf{K}$.

Partie 3 : Décodage via l'interpolation et l'algorithme d'Euclide étendu

On rappelle que, pour tous A et $B \in \mathbf{K}[X]$, non tous deux nuls, un polynôme $D \in \mathbf{K}[X]$ est un pgcd (plus grand commun diviseur) de A et B s'il satisfait aux deux conditions :

– D divise A et B ;

– si $E \in \mathbf{K}[X]$ divise A et B, alors E divise D.

Un pgcd de deux polynômes est donc défini à une constante multiplicative de $\mathbf{K}$ près. On notera $\mathrm{pgcd}(A, B)$ le pgcd unitaire de A et B.

Algorithme 4 Algorithme d'Euclide étendu.

Entrée : $A = \sum_{0 \leqslant i \leqslant n} a_i X^i$ et $B = \sum_{0 \leqslant i \leqslant m} b_i X^i \in \mathbf{K}[X]$ tels que $AB \neq 0$

Sortie : Un entier $\ell \in \mathbb{N}$, quatre suites finies $(Q_i)_{1 \leqslant i \leqslant \ell}$, $(R_i)_{0 \leqslant i \leqslant \ell+1}$, $(S_i)_{0 \leqslant i \leqslant \ell+1}$, $(T_i)_{0 \leqslant i \leqslant \ell+1}$ d'éléments de $\mathbf{K}[X]$

1. $R_0 \longleftarrow A$, $S_0 \longleftarrow 1$, $T_0 \longleftarrow 0$,
   $R_1 \longleftarrow B$, $S_1 \longleftarrow 0$, $T_1 \longleftarrow 1$
2. $i \longleftarrow 1$,
   tant que $R_i \neq 0$, faire
3. $(Q_i, R_{i+1}) \longleftarrow$ Algorithme 3 appliqué à (R_{i-1}, R_i)
   $S_{i+1} \longleftarrow S_{i-1} - Q_i S_i$
   $T_{i+1} \longleftarrow T_{i-1} - Q_i T_i$
   $i \longleftarrow i + 1$
4. Renvoyer $\ell = i - 1$ et les suites $(Q_i)_{1 \leqslant i \leqslant \ell}, (R_i)_{0 \leqslant i \leqslant \ell+1}, (S_i)_{0 \leqslant i \leqslant \ell+1}, (T_i)_{0 \leqslant i \leqslant \ell+1} \in \mathbf{K}[X]$

L'algorithme d'Euclide étendu fournit notamment un pgcd des deux polynômes en entrée et les coefficients de Bézout correspondants, à savoir respectivement les polynômes R_ℓ, S_ℓ et T_ℓ, comme on va le voir.

On notera $\mathcal{M}_{2,1}(\mathbf{K}(X))$ le $\mathbf{K}$-espace vectoriel des matrices à deux lignes et une colonne à coefficients dans $\mathbf{K}(X)$ et $\mathcal{M}_{2,2}(\mathbf{K}(X))$ le $\mathbf{K}$-espace vectoriel et l'anneau des matrices à deux lignes et deux colonnes à coefficients dans $\mathbf{K}(X)$.

On reprend les notations de l'Algorithme 4 et l'on introduit les matrices :

$$\mathcal{U}_0 = \begin{pmatrix} S_0 & T_0 \\ S_1 & T_1 \end{pmatrix} \in \mathcal{M}_{2,2}(\mathbf{K}(X)), \mathcal{W}_i = \begin{pmatrix} 0 & 1 \\ 1 & -Q_i \end{pmatrix} \in \mathcal{M}_{2,2}(\mathbf{K}(X)) \text{ pour tout } 1 \leqslant i \leqslant \ell$$

et

$$\mathcal{U}_i = \mathcal{W}_i \dots \mathcal{W}_1 \mathcal{U}_0 \in \mathcal{M}_{2,2}(\mathbf{K}(X)) \text{ pour tout } 0 \leqslant i \leqslant \ell.$$

Question 3.1. En reprenant les notations de l'Algorithme 4, démontrer que, pour tout $0 \leqslant i \leqslant \ell$, on a

(a). $\mathcal{U}_i \begin{pmatrix} A \\ B \end{pmatrix} = \begin{pmatrix} R_i \\ R_{i+1} \end{pmatrix}$,

(b). $\mathcal{U}_i = \begin{pmatrix} S_i & T_i \\ S_{i+1} & T_{i+1} \end{pmatrix}$ et $S_i A + T_i B = R_i$ (l'égalité est vraie aussi pour $i = \ell + 1$),

(c). $S_iT_{i+1} - S_{i+1}T_i = (-1)^i$,

(d). $\mathrm{pgcd}(A, B) = \mathrm{pgcd}(R_i, R_{i+1}) = R_\ell/\mathrm{cd}(R_\ell)$, où $\mathrm{cd}(R_\ell)$ désigne le coefficient dominant de R_ℓ.

Question 3.2. En reprenant les notations de l'Algorithme 4 et en supposant $\deg A \geqslant \deg B$, démontrer que

$$\deg S_i = \sum_{2 \leqslant j < i} \deg Q_j = \deg R_1 - \deg R_{i-1} \text{ pour tout } 2 \leqslant i \leqslant \ell + 1,$$

$$\deg T_i = \sum_{1 \leqslant j < i} \deg Q_j = \deg R_0 - \deg R_{i-1} \text{ pour tout } 1 \leqslant i \leqslant \ell + 1.$$

Question 3.3. Soit $n, m \in \mathbb{N}$, A et $B \in \mathbf{K}[X]$ avec $\deg A = n$ et $\deg B = m$. Démontrer que l'exécution de l'Algorithme 4 nécessite au plus $O(nm)$ opérations arithmétiques dans $\mathbf{K}$.

Question 3.4. Soit $M \in \mathbf{K}[X]$ de degré $n > 0$, $P \in \mathbf{K}[X]$ tel que $\deg P < n$. Soit $k \in \{0, \ldots, n\}$, déterminer un couple $(R, T) \in \mathbf{K}[X]^2$ tel que

$$T \neq 0 \text{ et } R \equiv TP \mod M, \deg R < k, \deg T \leqslant n - k. \tag{1}$$

On admet dans la suite du problème que l'évaluation d'un polynôme de $\mathbf{K}[X]$ de degré au plus t peut se faire en $O(t)$ opérations arithmétiques sur $\mathbf{K}$.

Algorithme 5

Entrée : n un entier naturel non nul, $u_0, u_1, \ldots, u_{n-1}$ distincts deux à deux, $v_0, v_1, \ldots, v_{n-1} \in \mathbf{K}$
Sortie : L'unique $P \in \mathbf{K}[X]$ tel que $\deg P \leqslant n - 1$ et $P(u_i) = v_i$ pour $i = 0, \ldots, n - 1$

1. $A \longleftarrow 1, P \longleftarrow 0$
2. Pour $i = 0, \ldots, n - 1$, faire $A \longleftarrow (X - u_i)A$
3. Pour $i = 0, \ldots, n - 1$, faire
4. $\quad A_i \longleftarrow$ quotient de la division euclidienne de A par $X - u_i$
   $\quad \alpha_i \longleftarrow A_i(u_i)$
   $\quad P \longleftarrow P + v_i A_i/\alpha_i$
5. Renvoyer P

Question 3.5. Soit n un entier naturel non nul, $u_0, u_1, \ldots, u_{n-1}, v_0, v_1, \ldots, v_{n-1} \in \mathbf{K}$. On suppose les u_i deux à deux distincts.

(a). Démontrer qu'un polynôme $P \in \mathbf{K}[X]$ tel que $\deg P \leqslant n - 1$ et $P(u_i) = v_i$ pour $i = 0, \ldots, n - 1$, est unique.

(b). Démontrer que l'Algorithme 5 calcule bien la sortie annoncée.

(c). Démontrer que le nombre d'opérations arithmétiques effectuées dans $\mathbf{K}$ pour calculer le polynôme P, sortie de l'Algorithme 5, est en $O(n^2)$.

Question 3.6. Soit n un entier naturel non nul, soit $k \in \{0, \dots, n\}$. Soit $u_0, \dots, u_{n-1} \in \mathbf{K}$ distincts deux à deux, $v_0, \dots, v_{n-1} \in \mathbf{K}$. On veut déterminer un couple $(R, T) \in \mathbf{K}[X]^2$ tel que

$$\begin{cases} R(u_i) = T(u_i)v_i \text{ pour tout } 0 \leqslant i < n, \\ \deg R < k, \\ T \neq 0 \text{ et } \deg T \leqslant n - k. \end{cases} \tag{2}$$

(a). Démontrer que résoudre (2) revient à résoudre (1) pour un couple (M, P) que l'on explicitera.

(b). En déduire une solution de (2).

Question 3.7. Démontrer que l'Algorithme 2 peut s'exécuter en un nombre $O(n^2)$ d'opérations arithmétiques sur $\mathbf{K}$.

Partie 4 : Borne de la programmation linéaire

On travaille dans toute cette partie avec des codes définis sur un alphabet $\mathcal{A}$ de cardinal $q \in \mathbb{N}$, $q \geqslant 2$. On se donne aussi un entier naturel non nul n.

On définit la taille d'un code comme le nombre de mots de ce code (précisons que le terme «taille» employé dans cette partie n'a pas de rapport avec celui employé dans la Partie 2). Soit C un code de longueur n et de taille M, on écrira que C est un (n, M)-code. Si de plus, C est de distance minimale d, on écrira que C est un (n, M, d)-code.

On notera $A_q(n, d)$ la plus grande taille possible d'un code sur $\mathcal{A}$ de longueur n et de distance minimale d, i.e.

$$A_q(n, d) = \max\{M;\ \text{il existe un } (n, M, d)\text{-code sur } \mathcal{A}\}.$$

Le but de cette partie est d'établir une borne sur $A_q(n, d)$ appelée borne de la programmation linéaire. Nous déduirons de cette borne une autre borne classique en théorie des codes.

Nous allons nous servir d'une certaine famille de polynômes (dans cette partie, on confondra polynôme et fonction polynomiale). Avant de la définir, nous allons généraliser la notion de coefficient binomial comme suit : si y et m sont deux nombres réels quelconques, on pose

$$\binom{y}{m} = \begin{cases} \dfrac{y(y-1)\cdots(y-m+1)}{m!} \text{ si } m \in \mathbb{N}, m \neq 0, \\ 1 \text{ si } m = 0, \\ 0 \text{ sinon.} \end{cases}$$

On vérifie aisément que cette définition prolonge bien celle rappelée dans le préambule.

Pour tout entier naturel k, le polynôme $K_k(x)$, où x est une variable de $\mathbb{R}$, est donné par

$$K_k(x) = \sum_{j=0}^{k} (-1)^j \binom{x}{j} \binom{n-x}{k-j} (q-1)^{k-j}.$$

On rappelle que pour tous u, $v \in \mathbb{R}$, avec $|u| < 1$, on a $(1+u)^v = \sum_{i=0}^{+\infty} \binom{v}{i} u^i$.

Question 4.1. Établir les propriétés suivantes.

(a). Soit $z \in \mathbb{R}$, $|z| < 1/(q-1)$, on a

$$\sum_{k=0}^{+\infty} K_k(x) z^k = (1+(q-1)z)^{n-x}(1-z)^x.$$

(b). On a, pour tout $k \in \mathbb{N}$,

$$K_k(x) = \sum_{j=0}^{k} (-1)^j q^{k-j} \binom{n-k+j}{j} \binom{n-x}{k-j}.$$

(c). Pour tout $k \in \mathbb{N}$, $K_k(x)$ est un polynôme de degré k, avec un coefficient dominant égal à $(-q)^k/k!$ et un terme constant $K_k(0) = \binom{n}{k}(q-1)^k$.

(d). Pour tous k, $\ell \in \mathbb{N}$, on a

$$\sum_{i=0}^{n} \binom{n}{i} (q-1)^i K_k(i) K_\ell(i) = \delta_{k\ell} \binom{n}{k} (q-1)^k q^n$$

où $\delta_{k\ell}$ est le symbole de Kronecker défini par $\delta_{k\ell} = 1$ si $k = \ell$ et 0 sinon. Indication : on pourra multiplier les deux membres de l'égalité par $y^k z^\ell$, avec y, $z \in]-1, 1[$, et sommer sur tous les $k, \ell \geqslant 0$, en expliquant pourquoi cette opération est licite.

(e). On a, pour tous k, $i \in \mathbb{N}$,

$$(q-1)^i \binom{n}{i} K_k(i) = (q-1)^k \binom{n}{k} K_i(k).$$

(f). On a, pour tous k, $\ell \in \mathbb{N}$,

$$\sum_{i=0}^{n} K_\ell(i) K_i(k) = \delta_{k\ell} q^n.$$

(g). On a, pour tout $m \in \mathbb{N}$,

$$\sum_{k=0}^{m} \binom{n-k}{n-m} K_k(x) = q^m \binom{n-x}{m}.$$

Indication : utiliser (b).

(h). Soit $r \in \mathbb{N}$, tout polynôme $f(x)$ de degré r peut s'exprimer sous la forme $f(x) = \sum_{k=0}^{r} f_k K_k(x)$ où $f_k = q^{-n} \sum_{i=0}^{n} f(i) K_i(k)$.

Soit C un (n, M)-code sur $\mathcal{A}$. On définit pour tout $i \in \mathbb{N}$, $0 \leqslant i \leqslant n$,

$$A_i(C) = \frac{1}{M} \mathrm{card}\{(u, v) \in C \times C : d(u, v) = i\}.$$

La suite finie $(A_i(C))_{i=0}^n$ est appelée la distribution des distances de C.

Remarque. La distribution des distances de C ne dépend pas de l'alphabet mais seulement du cardinal de l'alphabet. Nous supposerons donc à présent que l'alphabet $\mathcal{A}$ est l'anneau $\mathbb{Z}/q\mathbb{Z}$.

Comme dans la Partie 2, on note $w_H(x) = \delta_H(x, 0)$ pour tout $x \in (\mathbb{Z}/q\mathbb{Z})^n$.

Question 4.2. Soit ζ une racine primitive q-ième de l'unité dans $\mathbb{C}$, i.e. $\zeta^q = 1$ mais $\zeta^j \neq 1$ pour tout $j \in \mathbb{N}$, $1 \leqslant j \leqslant q-1$. On suppose que $\mathbf{u} \in (\mathbb{Z}/q\mathbb{Z})^n$ est un mot tel que $w_H(\mathbf{u}) = i$. Démontrer que

$$\sum_{\substack{\mathbf{v} \in (\mathbb{Z}/q\mathbb{Z})^n \\ w_H(\mathbf{v}) = k}} \zeta^{\mathbf{u}\cdot\mathbf{v}} = K_k(i)$$

où, si $\mathbf{u} = (u_1, \ldots, u_n)$ et $\mathbf{v} = (v_1, \ldots, v_n)$, on a $\mathbf{u} \cdot \mathbf{v} = \sum_{j=1}^n u_j v_j$ (on confond u_j et v_j avec leurs représentants dans $\{0, \ldots, q-1\}$: c'est bien défini puisque $\zeta^q = 1$).

Question 4.3. Soit C un code de longueur n sur l'alphabet $\mathbb{Z}/q\mathbb{Z}$. Démontrer que

$$\sum_{i=0}^{n} A_i(C) K_k(i) \geqslant 0$$

pour tout $k \in \mathbb{N}$, $0 \leqslant k \leqslant n$.

Question 4.4. *Borne de la programmation linéaire (première version).* Pour tous entiers n et d tels que $1 \leqslant d \leqslant n$, démontrer que

$$A_q(n,d) \leqslant \max\left\{ \sum_{i=0}^{n} A_i ; A_0 = 1, A_i = 0 \text{ pour } 1 \leqslant i < d, A_i \geqslant 0 \text{ pour } 0 \leqslant i \leqslant n, \right.$$
$$\left. \sum_{i=0}^{n} A_i K_k(i) \geqslant 0 \text{ pour } 0 \leqslant k \leqslant n \right\}.$$

Question 4.5. *Borne de la programmation linéaire (seconde version).* Pour tous entiers n et d tels que $1 \leqslant d \leqslant n$, soit $f(x) = 1 + \sum_{k=1}^{n} f_k K_k(x)$ un polynôme tel que $f_k \geqslant 0$ pour tout $1 \leqslant k \leqslant n$ et $f(i) \leqslant 0$ pour $d \leqslant i \leqslant n$. Démontrer que $A_q(n,d) \leqslant f(0)$.

Question 4.6. *Borne de Singleton.* Déduire de la question précédente que pour tous entiers n et d tels que $1 \leqslant d \leqslant n$, on a $A_q(n,d) \leqslant q^{n-d+1}$.

Remarque. Il s'agit bien de la même borne que dans la Partie 2 : dans la pratique, les codes $\mathrm{RS}_{\alpha,k}$ sont considérés sur des corps de cardinal q, où q est une puissance d'un nombre premier p.

* *
*