Feuille d'exercices : Correction 5

LFSR (2) - Algorithme de Berlekamp-Massey

Matrices de Hankel. Les *matrices de Hankel* d'une suite (s_n) d'élements de $\mathbb{F}_q$ sont les matrices de $\mathscr{M}_{m,n}(\mathbb{F}_q)$ suivantes :

$$H_{j,m,n} = \begin{pmatrix} s_j & s_{j+1} & \cdots & s_{j+n-1} \\ s_{j+1} & s_{j+2} & \cdots & s_{j+n} \\ \vdots & \vdots & & \vdots \\ s_{j+m-1} & s_{j+m} & \cdots & s_{j+m+n-2} \end{pmatrix} \quad H_{j,m} = \begin{pmatrix} s_j & s_{j+1} & \cdots & s_{j+m-1} \\ s_{j+1} & s_{j+2} & \cdots & s_{j+m} \\ \vdots & \vdots & & \vdots \\ s_{j+m-1} & s_{j+m} & \cdots & s_{j+2m-2} \end{pmatrix}$$

La transposée de $H_{j,m,n}$ est $H_{j,n,m}$. Les matrices carrées, notées $H_{j,m,m} = H_{j,m}$, sont symétriques. Le premier exercice énonce quelques résultats simples sur le rapport entre matrices de Hankel et LFSR.

Exercice 1 (LFSR et matrices de Hankel).

1. Soit A la la matrice d'un LFSR de taille m qui engendre (s_n). Montrer que

$$\text{pour tout entier } k,\ A \cdot H_{j,m,k} = H_{j+1,m,k}\ .$$

Solution : Colonne par colonne, c'est la relation de récurrence exprimée matriciellement de j à $j+k-1$.

2. Montrer que $P(X) = \sum_{i=0}^{m} c_i X^i$ $(c_0 = 1)$ est le polynôme de connexion d'un LFSR de taille m qui engendre une suite dont les $n+1$ premiers termes sont $(s_0, \ldots, s_n)$ si et seulement si :

$$H_{0,n-m+1,m+1} \begin{pmatrix} c_m \\ \vdots \\ c_0 \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Solution : C'est la relation de récurrence vérifiée $n-m$ fois.

3. Montrer que la suite (s_n) est engendrée par un LFSR de taille m, de polynôme de connexion $\sum_{i=0}^{m} c_i X^i$ $(c_0 = 1)$ si et seulement si pour tout j on a l'égalité suivante (ou sa transposée) :

$$H_{j,m+1} \begin{pmatrix} c_m \\ \vdots \\ c_0 \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Solution : Colonne par colonne, c'est la relation de récurrence, de $j+m$ à $j+2m$.

Ces trois propriétés peuvent aussi s'écrire sous forme transposée.

Exercice 2. Soit une suite (s_n) d'élements de $\mathbb{F}_q$ engendrée par une récurrence linéaire d'ordre m, de matrice A, de polynôme de connexion $\chi^*(X) = \sum_{i=0}^{m} c_i X^i$ $(x_0 = 1)$ et d'initialisation non nulle $(s_0, \ldots, s_{m-1}) \neq (0, \ldots, 0)$.

1. Montrer qu'aucun LFSR de taille plus petite n'engendre cette suite si et seulement si la matrice $H_{0,m}$ est inversible.

Solution : D'après le résultat de l'exercice 2 si un LFSR de taille $l < m$ de polynôme de connexion $\sum_{i=0}^{l} b_i X^i$ $(b_0 = 1)$, engendre la suite s_n, alors :

$$\underbrace{(0, \ldots, 0}_{m-l+1}, b_l, \ldots, b_0) \cdot H_{0,m} = (0, \cdots, 0)$$

et donc $H_{0,m}$ n'est pas inversible.
Réciproquement si $H_{0,m}$ n'est pas inversible, alors soit $(b_{m-1}, \ldots, b_0)$ tels que :

$$H_{0,m} \begin{pmatrix} b_{m-1} \\ \vdots \\ b_0 \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$$

par multiplication par A :

$$H_{j,m}\begin{pmatrix} b_{m-1} \\ \vdots \\ b_0 \end{pmatrix} = A^j H_{0,m}\begin{pmatrix} b_{m-1} \\ \vdots \\ b_0 \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$$

c'est-à-dire que l'on a $\sum_{i=0}^{m-1} b_i s_{j+m-1-i} = 0$, pour tout entier naturel j, soit une relation de récurrence d'ordre strictement inférieur à m.

2. On suppose de plus que $c_m \neq 0$. Montrer que pour un entier j quelconque, le LFSR est de taille minimale si et seulement si $H_{j,m}$ est inversible.

Solution : Si $c_m \neq 0$, la matrice A est inversible, donc d'après l'exercice 1 (en prenant $k = m$) $H_{j,m}$ est inversible si et seulement si $H_{0,m}$ est inversible.

3. En déduire que, si une suite est engendrée par un LFSR de taille m minimale, alors un LFSR qui engendre cette suite, est entièrement déterminé par les $2m$ premiers termes de la suite, par $2m$ termes consécutifs si le LFSR est de matrice inversible.

Solution : Les matrices $H_{0,m}$ et $H_{1,m}$ sont déterminées par les $2m$ premiers termes de la suite. Comme le LFSR qui engendre la suite est de taille minimale, $H_{0,m}$ est inversible. Les coefficients du LFSR sont donnés par la matrice de celui-ci $A\cdot = H_{1,m} \cdot H_{0,m}^{-1}$

Sachant qu'une suite est produite par un LFSR propre (matrice inversible) de taille m minimale, on a donc par pivot de Gauss un algorithme en temps cubique (en fonction de m) pour déterminer les coefficients et l'initialisation du LFSR à partir de $2m$ termes consécutifs de la suite engendrée. On va voir un algorithme plus efficace en temps quadratique.

Complexité linéaire. La *complexité linéaire d'une suite infinie* $s = (s_n)_{n\in\mathbb{N}}$ d'éléments de $\mathbb{F}_q$, notée $\Lambda(s)$, est :

— $\Lambda(s) = 0$, si la suite est constante égale à 0 ;
— $\Lambda(s) = m$, si la suite est engendré par un LFSR, où m est la longueur du plus petit LFSR qui engendre la suite s ;
— $\Lambda(s) = \infty$, si la suite s n'est pas engendrée par un LFSR.

Si (s_n) est périodique, et si sa série génératrice se met sous forme de fraction rationnelle irréductible $\sum_{n\geq 0} s_n X^n = \frac{G(X)}{P(X)}$, alors sa complexité linéaire est $\Lambda((s_n)) = \sup\big(\deg(G)+1, \deg(P)\big)$.
La *complexité linéaire d'une suite finie* $s^N = (s_0, \ldots, s_{N-1})$ d'éléments de $\mathbb{F}_q$ est la longueur minimale d'un LFSR qui engendre une suite infinie s dont les premiers termes sont $(s_0, \ldots, s_{N-1})$.

Exercice 3. On note $s^N = (s_0, \ldots, s_{N-1})$ une suite finie de longueur N, $s = (s_n)_{n\in\mathbb{N}}$ une suite infinie. Montrer que :

1. pour $N \geq 1$, $0 \leq \Lambda(s^N) \leq N$;
2. Si s est périodique de période de longueur p, alors $\Lambda(s) \leq p$;
3. $\Lambda(s^N) = 0$ si et seulement si s^N est identiquement nulle ;
4. $\Lambda(s^N) = N$ si et seulement si $s^N = (0, \ldots, 0, x)$ où $x \neq 0$.

Exercice 4.

1. Soient deux suites (s_n) et (t_n) sur $\mathbb{F}_q$. Montrer, en utilisant leurs séries génératrices, que :

$$\Lambda((s_n + t_n)) \leq \Lambda((s_n)) + \Lambda((t_n)) \ .$$

Solution : Chacune des deux séries est donnée par une fraction rationnelle, soient $\frac{G(X)}{P(X)}$ et $\frac{H(X)}{Q(X)}$. On a $\Lambda((s_n)) = \sup(\deg(G)+1, \deg(P))$, $\Lambda((t_n)) = \sup(\deg(H)+1, \deg(>))$. La suite $(s_n + t_n)$ a pour série génératrice $\frac{G(X)Q(X)+P(X)H(X)}{P(X)Q(X)}$; On a bien $\deg(P(X)Q(X)) = \deg(P) + \deg(Q) \leq \Lambda((s_n)) + \Lambda((t_n))$, $\deg(G(X)Q(X)) < \Lambda((s_n)) + \Lambda((t_n))$, $\deg(P(X)H(X)) < \Lambda((s_n)) + \Lambda((t_n))$.
comme une fraction rationnelle de dénominateur le produit donc de degré la somme inférieur ou égal à $\Lambda((s_n)) + \Lambda((t_n))$.

2. En déduire la même propriété pour deux suites finies de même longueur N $s^N = (s_0, \ldots, s_{N-1})$ et $t^N = (t_0, \ldots, t_{N-1})$, c'est-à-dire qu'en posant $(s+t)^N = (s_0 + t_0, \ldots, s_{N-1} + t_{N-1})$, on a :

$$\Lambda\left((s_n + t_n)^N\right) \leq \Lambda(s^N) + \Lambda(t^N) \ .$$

Solution : On prolonge les suites s^N et t^N en prenant pour chacune d'entre elle un LFSR de taille minimale. La question précédente fournit un LFSR de taille inférieure ou égale à $\Lambda((s_n)) + \Lambda((t_n))$, qui engendre en particulier $(s+t)^N$

Profil de complexité linéaire. La complexité linéaire n'est pas une bonne mesure de la complexité réelle d'une suite finie, du fait que si celle-ci est nulle sauf en son dernier élément, la complexité est maximale (exercice 3, question 4). Une mesure plus significative est le *profil de complexité linéaire* d'une suite (finie ou infinie), soit la suite des complexités linéaires des suites partielles :

— si $s = (s_n)$, et $\lambda_n = \Lambda(s^n)$, le profil de complexité linéaire de s est la suite $(\lambda_1, \ldots, \lambda_n, \ldots)$.
— si $s^N = (s_0, \ldots, s_{N-1})$, le profile de complexité linéaire de s^N est la suite $(\lambda_1, \ldots, \lambda_N)$.

On va montrer que le profil de complexité linéaire d'une suite (s_n) a les propriétés suivantes :

i. Si $j > i$ alors $\lambda_j \geq \lambda_i$;

ii. Si $\lambda_{j+1} > \lambda_j$, alors $\lambda_j + \lambda_{j+1} = j+1$;

iii. Si $\lambda_{j+1} > \lambda_j$, alors $\lambda_j \leq \frac{j}{2}$;

On recherche une suite $s = (s_n)$ de profil de complexité linéaire qui ne fait pas de sauts importants, et qui donc, au vu de ces propriétés, sera proche de la droite $\lambda = \frac{j}{2}$.

Exercice 5. Montrer la propriété **i**, et que la propriété **iii** se déduit de **ii** (la propriété **ii** sera démontrée à l'exercice 7).

Solution : Immédiat.

Exercice 6. **1.** On suppose qu'un LFSR de taille λ_n engendre $(s_0, \ldots, s_{n-1})$ et n'engendre pas $(s_0, \ldots, s_n)$. Appelons (t_n) la suite engendrée par ce LFSR. Décrire les $n+1$ premiers termes de la suite $(s_n - t_n)$ et déduire des exercices 4 et 3, que sous cette hypothèse, $\lambda_n + \lambda_{n+1} \geq n+1$.

Solution : La suite $(s-t)^{n+1}$ des $n+1$ premiers termes de $(s_n - t_n)$ s'écrit $(0, \ldots, 0, a)$ avec $a \neq 0$, et donc $\Lambda((s-t)^{n+1}) = n+1$ (exercice 3, question 4), donc $\Lambda(s^{n+1}) + \Lambda(t^{n+1}) \geq n+1$, soit $\lambda_{n+1} + \lambda_n \leq n+1$,

2. En déduire que la propriété **ii** ci-dessus équivaut à la propriété **iv** :

iv. si un LFSR de taille λ_j qui engendre $(s_0, \ldots, s_{j-1})$ n'engendre pas $(s_0, \ldots, s_j)$, alors $\lambda_{j+1} = \sup(\lambda_j, j+1-\lambda_j)$.

Solution : Supposons qu'un LFSR de taille λ_j qui engendre $(s_0, \ldots, s_{j-1})$ n'engendre pas $(s_0, \ldots, s_j)$. De **ii** et de la question précédente on déduit que $\lambda_{j+1} = \sup(\lambda_j, j+1-\lambda_j)$. De **iv** on déduit immédiatement **ii**.

Algorithme de Berlekamp-Massey. L'algorithme de Berlekamp-Massey est un algorithme de complexité quadratique qui permet de calculer un LFSR de taille minimale qui engendre une suite finie, ainsi que le profil de complexité linéaire de cette suite finie.

Exercice 7. Le but de l'exercice est de montrer la propriété **ii** (et donc **iv**) par récurrence pour les sous-suites finies des premiers éléments d'une suite finie (s_n), en montrant comment calculer les polynômes minimaux correspondants (les polynômes de connexion correspondant aux λ_j successifs).

La démonstration utilise le résultat de l'exercice 6. Le principe en est le suivant. Rappelons qu'un LFSR est déterminé par sa taille et son polynôme de connexion.

Supposons un LFSR de taille minimale λ_n déterminé à l'étape n, c'est-à-dire qu'il en engendre la suite jusqu'à s_{n-1}. À l'étape $n+1$, si le même LFSR convient on a terminé. Sinon on cherche un LFSR de même taille qui engendre la suite jusqu'à s_n, et qui sera donc forcément de taille minimale, ce qui d'après l'exercice 6 demande $2\lambda_n \geq n+1$. On montre, en exhibant dans ce cas un tel LFSR de taille $\lambda_{n+1} = \lambda_n$, que cette dernière condition est suffisante.

Sinon, c'est donc que $2\lambda_n < n+1$, et on sait toujours d'après l'exercice 6 que $\lambda_{n+1} \geq n+1-\lambda_n$, on cherche alors un LFSR de taille $\lambda_{n+1} = n+1-\lambda_n$ qui engendre la suite jusqu'à s_n, et qui sera forcément de taille minimale. Pour l'exhiber, on utilise que la propriété **ii** est démontrée pour un

certain m, $m \leq n$, et on montre alors la propriété ii pour n, d'où une démonstration par récurrence sur n.

Soit P_n le polynôme de connexion du LFSR de taille λ_n obtenu à l'étape n, qui engendre donc la suite finie $(s_0, \ldots, s_{n-1})$. On note ses coefficients ainsi :

$$P_n(X) = \sum_{i=0}^{\lambda_n} c_{n,i} X^i \quad (c_{n,0} = 1).$$

Le *coefficient critique* d'ordre n est d_n défini par :

$$d_n = \sum_{i=0}^{\lambda_n} s_{n-i} c_{n,i} \ .$$

1. On suppose que pour tout $k < n$, si $\lambda_{k+1} > \lambda_k$, alors $\lambda_{k+1} + \lambda_k = k+1$. On suppose de plus qu'il existe un $k < n$ tel que $\lambda_{k+1} > \lambda_k$, et on appelle m le plus grand $k < n$ tel que $\lambda_{k+1} > \lambda_k$.

   a. Vérifier que $d_m \neq 0$ et $\lambda_n = m + 1 - \lambda_m$.

   Solution : Par définition de m, $d_m \neq 0$, $\lambda_{m+1} > \lambda_m$ et $\lambda_n = \lambda_{m+1}$. Comme

   b. Montrer que $H_{0,n-\lambda_n+1,\lambda_n+1} \begin{pmatrix} c_{n,\lambda_n} \\ \vdots \\ . \\ c_{n,0} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ d_n \end{pmatrix}$.

   Solution :

$$H_{0,n-\lambda_n+1,\lambda_n+1} = \begin{pmatrix} s_0 & s_1 & \cdots & s_{\lambda_n} \\ s_1 & s_2 & \cdots & s_{\lambda_n+1} \\ \vdots & \vdots & & \vdots \\ s_{n-\lambda_n} & s_{n-\lambda_n+1} & \cdots & s_n \end{pmatrix}$$

   Les lignes de 0 à $(n-1)-\lambda_n$ expriment que le LFSR de taille λ_n obtenu à l'étape n engendre la suite $s_0, \ldots, s_{n-1}$. La dernière ligne donne bien le calcul du coefficient critique.

   c. On suppose $d_n = 0$. Montrer que $(s_0, \ldots, s_n)$, est engendré par un LFSR de taille minimale $\lambda_{n+1} = \lambda_n$, complexité linéaire de cette suite, et de polynôme de connexion $P_{n+1} = P_n$.

   d. On suppose que $d_n \neq 0$ et $2\lambda_n \geq n+1$. Montrer que $m - \lambda_m \geq n - \lambda_n$. En déduire un vecteur colonne V de taille $\lambda_n + 1$ utilisant les coefficients de P_m tel que :

$$H_{0,n-\lambda_n+1,\lambda_n+1} \cdot V = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ d_m \end{pmatrix} .$$

   Solution :

$$H_{0,n-\lambda_n+1,\lambda_n+1} = \begin{pmatrix} s_0 & \cdots & s_{(m-\lambda_m)-(n-\lambda_n)} & \cdots & s_{m-(n-\lambda_n)} & \cdots & s_{\lambda_n} \\ \vdots & & \vdots & & \vdots & & \vdots \\ s_{n-\lambda_n} & \cdots & s_{m-\lambda_m} & \cdots & s_m & \cdots & s_n \end{pmatrix} ; \ V = \begin{pmatrix} 0 \\ \vdots \\ c^m_{\lambda_m} \\ \vdots \\ c^m_0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} .$$

   e. Conclure, qu'avec les hypothèses de la question précédente, $R = P_n - \frac{d_n}{d_m} X^{n-m} P_m$ est le polynôme de connexion d'un LFSR de longueur λ_n qui engendre $(s_0, \ldots, s_n)$, puis que $\lambda_{n+1} = \lambda_n$, et que l'on peut choisir $P_{n+1} = R$.

Solution : On a :

$$H_{0,n-\lambda_n+1,\lambda_n+1}\cdot\left(\begin{pmatrix}c_{n,\lambda_n}\\ \vdots\\ c_{n,0}\end{pmatrix}-\frac{d_m}{d_n}V\right)==\begin{pmatrix}0\\ \vdots\\ 0\end{pmatrix}$$

et le polynôme correspondant est bien R.

f. On suppose maintenant que $d_n\neq 0$ et $2\lambda_n<n+1$. En déduire que $\lambda_{n+1}>\lambda_n$ et $\lambda_{n+1}\geq n+1-\lambda_n$. En procédant de façon analogue aux deux questions précédentes, montrer que le même polynôme $R=P_n-\frac{d_n}{d_m}X^{n-m}P_m$ est le polynôme de connexion d'un LFSR de longueur $n+1-\lambda_n$ qui engendre $(s_0,\dots,s_n)$. En déduire que $\lambda_{n+1}=n+1-\lambda_n$, et que l'on peut choisir $P_{n+1}=R$.

Solution : On pose $l=n+1-\lambda_n$. On a $l>\lambda_n$. D'autre part $\lambda_n=n+1-l=m+1-\lambda_m$ et $n-l=m-\lambda_m$. On a $(l>\lambda_m)$:

$$H_{0,n-l+1,l+1}=\begin{pmatrix}s_0 & \cdots & s_{\lambda_m} & \cdots & s_l\\ \vdots & & \vdots & & \vdots\\ s_{m-\lambda_m} & \cdots & s_m & \cdots & s_n\end{pmatrix};\ H_{0,n-l+1,l+1}\cdot\begin{pmatrix}0\\ \vdots\\ 0\\ c_{n,\lambda_n}\\ \vdots\\ c_{n,0}\end{pmatrix}=\begin{pmatrix}0\\ \vdots\\ .\\ .\\ \vdots\\ 0\\ d_n\end{pmatrix}$$

et

$$H_{0,n-\lambda_{n+1}+1,\lambda_{n+1}+1}\cdot\begin{pmatrix}c_{m,\lambda_m}\\ \vdots\\ c_{m,0}\\ 0\\ \vdots\\ 0\end{pmatrix}=\begin{pmatrix}0\\ \vdots\\ .\\ \vdots\\ 0\\ d_m\end{pmatrix};\ H_{0,n-\lambda_{n+1}+1,\lambda_{n+1}+1}\cdot\left(\begin{pmatrix}0\\ \vdots\\ 0\\ c_{n,\lambda_n}\\ \vdots\\ c_{n,0}\end{pmatrix}-\frac{d_n}{d_m}\begin{pmatrix}c_{m,\lambda_m}\\ \vdots\\ c_{m,0}\\ 0\\ \vdots\\ 0\end{pmatrix}\right)=\begin{pmatrix}0\\ \vdots\\ .\\ \vdots\\ .\\ 0\end{pmatrix}.$$

d'où un LFSR de taille $l=n+1-\lambda_n$ qui engendre $(s_0,\dots,s_n)$ donc $\lambda_{n+1}\geq n+1-\lambda_n$, et d'après l'exercice 6, question 1, $\lambda_{n+1}=l=n+1-\lambda_n$, et le polynôme de connexion correspondant est $P_{n+1}=P_n-\frac{d_n}{d_m}X^{n-m}P_m$.

2. La complexité linéaire de la suite constante nulle (suite vide y compris) est 0 et son polynôme minimal est le polynôme constant égal à 1. Vérifiez que pour la suite $(\underbrace{0,\dots,0}_{p},a)$, $p\geq 0$, $x\neq 0$, le polynôme $1-aX^{p+1}$ convient (y compris si $p=0$).

3. Montrer par récurrence la propriété ii.

4. Donner une définition par récurrence de (P_n,λ_n).

Solution :

— Si (s_n) est identiquement nulle, $P_n=1$, $\lambda_n=0$;

— Pour $s_p=a$ le premier terme non nul de la suite :

$$P_n=1 \text{ et } \lambda_n=0 \text{ pour } n<p\ ;\ P_p=1-aX^{p+1} \text{ et } \lambda_p=p+1\ ;$$

— Pour $n+1>p$, soit m le plus grand $k<n$ tel que $\lambda_{m+1}>\lambda_m$ (existe car p est un tel entier) :

— si $d_{n+1}=0$, $P_{n+1}=Pn$, $\lambda_{n+1}=\lambda_n(=\lambda_{m+1})$;

— si $d_{n+1}\neq 0$, $P_{n+1}(X)=P_n(X)+X^{n-m}P_m(X)$ et $\lambda_{n+1}=\sup(\lambda_n,n+1-\lambda_n)$.

Exercice 8 (Algorithme de Berlekamp-Massey). 1. En utilisant l'exercice précédent, montrer que l'algorithme suivant calcule la complexité linéaire et le polynôme de connexion d'un LFSR engendrant une suite d'éléments de $\mathbb{F}_2$:

Entrée : $s^n=(s_0,\dots,s_{n-1})$

Sortie : $\lambda=\Lambda(s^n)$, P polynôme de connexion associé.

$P(X):=1$, $Q(X):=1$, $\lambda:=0$, $m:=-1$, d, $T(X)$ /* *Initialisation* */

`for` $k=0$ `to` $n-1$ `do`

$d:=\sum_{i=0}^{\lambda}c_i s_{k-i}$ /* $P=\sum_{i=0}^{\lambda}c_iX^i$ */

```
    if d ≠ 0 then
            T(X) := P(X)
            P(X) := P(X) + Q(X)X^{k−m}
            if 2·λ ≤ k then
                    λ := k + 1 − λ
                    m := k
                    Q(X) := T(X)
return λ, P
```

2. Montrer que l'algorithme de Berlekamp-Massey est en temps $O(n^2)$.

Exercice 9. Le tableau suivant détaille les calculs successifs de l'algorithme de Berlekamp-Massey pour la suite en entrée indiquée sur la dernière colonne :

k	d	P	λ	Q	m	T	s
		1	0	1	-1		
0	1	$X+1$	1	1	0	1	1
1	0	$X+1$	1	1	0	1	1
2	1	X^2+X+1	2	$X+1$	2	$X+1$	0
3	1	1	2	$X+1$	2	X^2+X+1	0
4	1	X^3+X^2+1	3	1	4	1	1
5	0	X^3+X^2+1	3	1	4	1	0
6	0	X^3+X^2+1	3	1	4	1	1
7	0	X^3+X^2+1	3	1	4	1	1
8	1	$X^4+X^3+X^2+1$	6	X^3+X^2+1	8	X^3+X^2+1	0
9	1	X^2+X+1	6	X^3+X^2+1	8	$X^4+X^3+X^2+1$	1
10	1	X^5+X^4+X+1	6	X^3+X^2+1	8	X^2+X+1	0
11	1	$X^6+X^4+X^3+X+1$	6	X^3+X^2+1	8	X^5+X^4+X+1	1

On peut programmer cet algorithme en C en utilisant des entiers (ou des tableaux d'entiers si la taille le nécessite) pour représenter les suites et les polynômes.

1. Programmer l'algorithme pour des suites de longueur inférieure à 64 (un entier machine suffit pour les représenter). On peut aligner les coefficients du polynôme avec ceux de la suite pour que le calcul de d se fasse par un produit bit à bit.

   D'une façon ou d'une autre, vous devez assurer la compatiblité entre votre programme pour les LFSR, et celui-ci (ordre des bits de la suite, et du polynôme).

2. tester le programme : produire par exemple un LFSR de taille 16 de période maximale (à l'aide du programme déjà écrit sur les LFSR), engendrer les 32 premiers bits, et vérifier que l'algorithme de Berlekamp-Massey pour ces 32 bits donne bien le LFSR d'origine.