Université Paris 8 Année 2022–2023

M2 Mathématiques et applications, parcours ACC

Algorithmes arithmétiques II – Solutions feuille de TD 2

29/09/2022

Retrouvez le sujet du TD et d'autres exercices à l'adresse :

`www.math.univ-paris13.fr/~lavauzelle/teaching/2022-23/algorithmes-arithmetiques.html`

($\star$) exercice fondamental ($\star\star$) pour s'entraîner ($\star\star\star$) pour aller plus loin 💻 sur machine

Exercice 1. ($\star$) Suite vectorielle et LFSR.

Soit $\boldsymbol{u} = (u_n)_{n\in\mathbb{N}}$ une suite récurrente linéaire scalaire non-nulle, produite par un LFSR de dimension L. On note $P(X) = \sum_{j=0}^{d} c_j X^j$ le polynôme de connexion du LFSR. On rappelle qu'on a donc

$$\sum_{j=0}^{d} c_j u_{n-j} = 0, \quad c_0 = 1\,.$$

Soit maintenant $\boldsymbol{v} = (\boldsymbol{v}_k)$ une suite définie par

$$\forall k \in \mathbb{N}, \quad \boldsymbol{v}_k := (u_k, u_{k+1}, \ldots, u_{k+L-1})^\top \in \mathbb{F}^L\,.$$

Question 1.– Démontrer que $\boldsymbol{v}$ est une suite récurrente linéaire sur $\mathbb{F}^L$. En donner une description sous forme de suite itérée, dont on précisera la matrice $\boldsymbol{A}$.

Question 2.– Que vaut $\det(\boldsymbol{A})$? En déduire une condition suffisante pour que la suite $\boldsymbol{u}$ ne soit pas nulle à partir d'un certain rang.

Question 3.– À l'aide des questions précédentes, démontrer que si le corps $\mathbb{F}$ est de cardinal q fini (*i.e.* $\mathbb{F} = \mathbb{F}_q$), alors la suite $\boldsymbol{u}$ a une période $\leq q^L - 1$.

Solutions de l'Exercice 1.

Solution Q1. On a :

$$\boldsymbol{v}_k = \begin{pmatrix} u_k \\ u_{k+1} \\ \vdots \\ u_{k+L-1} \end{pmatrix} = \begin{pmatrix} -\sum_{j=1}^{d} c_j u_{k-j} \\ -\sum_{j=1}^{d} c_j u_{k+1-j} \\ \vdots \\ -\sum_{j=1}^{d} c_j u_{k+L-1-j} \end{pmatrix} = -\sum_{j=1}^{d} c_j \begin{pmatrix} u_{k-j} \\ u_{k+1-j} \\ \vdots \\ u_{k+L-1-j} \end{pmatrix} = -\sum_{j=1}^{d} c_j \boldsymbol{v}_{k-j}$$

On peut donc écrire $\boldsymbol{v}_{k+1} = \boldsymbol{A} \cdot \boldsymbol{v}_k$ avec :

$$\boldsymbol{A} = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ & 0 & 1 & \ddots & \\ & & \ddots & \ddots & 0 \\ 0 & & & 0 & 1 \\ -c_{L-1} & -c_{L-2} & \cdots & \cdots & -c_0 \end{pmatrix}$$

Puis

$$\boldsymbol{v}_k = \boldsymbol{A}^k \cdot \boldsymbol{v}_0$$

La matrice $\boldsymbol{A}$ est appelée *matrice compagnon* du polynôme P.

Solution Q2. Notons d'abord qu'il est équivalent de dire $\boldsymbol{u}$ et $\boldsymbol{v}$ sont nulles à partir d'un certain rang.

On a $\det \boldsymbol{A} = \pm c_{L-1}$. Donc $\boldsymbol{v}_k = \boldsymbol{A}^k \boldsymbol{v}_0 \neq \boldsymbol{0}$ dès lors que $\boldsymbol{v}_0 \neq \boldsymbol{0}$ et $c_{L-1} \neq 0$. Cette dernière condition est équivalente à dire que le polynôme de connexion $\deg P$ a un degré maximal.

Solution Q3. Le vecteur $\boldsymbol{v}_k \in \mathbb{F}^L$ peut prendre, au maximum, q^L valeurs possibles. S'il existe un indice k tel que $\boldsymbol{v}_k = \boldsymbol{0}$, alors pour tout $k' \geq k$, on a $\boldsymbol{v}_{k'} = \boldsymbol{A}^{k'-k}\boldsymbol{v}_k = \boldsymbol{0}$. Dans ce cas $\boldsymbol{u}$ a période 1.

Sinon, il existe k, k' distants d'au plus $q^L - 1$ tels que $\boldsymbol{v}_k = \boldsymbol{v}'_k$. On a alors, pour tout $t \geq 0$:

$$\boldsymbol{v}_{k+t} = \boldsymbol{A}^t \boldsymbol{v}_k = \boldsymbol{A}^t \boldsymbol{v}_{k'} = \boldsymbol{v}_{k'+t}$$

Donc la période de $\boldsymbol{v}$ (et a fortiori de $\boldsymbol{u}$) est d'au plus $q^L - 1$.

Exercice 2. ($\star\star$) Calcul du polynôme de connexion par l'algorithme d'Euclide.

Dans cet exercice, on étudie une méthode permettant de calculer le polynôme de connexion minimal d'une suite scalaire $\boldsymbol{b} \in \mathbb{F}^{\mathbb{N}}$, en utilisant l'algorithme d'Euclide étendu.

Pour cela, on considère les $2n$ premiers termes de la suite $\boldsymbol{b}$, on note $B(X) = \sum_{i=0}^{2n-1} b_i X^i \in \mathbb{F}[X]$, et on cherche donc un polynôme de connexion $P(X)$ de degré $\leq n$ tel que $P(X)B(X) \mod X^{2n}$ est de degré $< n$.

Question 1.– Rappeler l'algorithme d'Euclide étendu pour les polynômes.

Question 2.– Exemple : supposons que $\mathbb{F} = \mathbb{F}_2$, $n = 4$ et $\boldsymbol{b} = (0, 1, 1, 1, 0, 0, 1, 0)$.

1. Donner l'expression du polynôme $B(X)$.
2. Déterminer un polynôme de connexion de la suite, de degré < 4. On pourra, si besoin, utiliser l'algorithme de Berlekamp–Massey.
3. Appliquer l'algorithme d'Euclide étendu à $A(X) = X^8$ et $B(X)$.
4. Commenter les résultats obtenus.

Dans le cas général, on exécute l'algorithme d'Euclide étendu sur les entrées $A(X) = X^{2n}$ et $B(X) = \sum_{i=0}^{2n-1} b_i X^i$. Les polynômes successivement calculés par l'algorithme sont notés R_i, Q_i, U_i, V_i et satisfont :

$$R_{i-1} = Q_i R_i + R_{i+1}, \quad U_{i-1} = Q_i U_i + U_{i+1}, \quad V_{i-1} = Q_i V_i + V_{i+1}$$

avec initialement $R_0 = A$ et $R_1 = B$.

On note $k \geq 1$ le premier indice pour lequel le reste $R_k(X)$ a degré $< n$. Autrement dit, on a également $\deg R_{k-1} \geq n$.

Question 3.– Quelle est la relation entre $\deg V_k$, $\deg R_{k-1}$ et $\deg A$?

Question 4.– Démontrer que V_k est un polynôme de connexion de la suite $\boldsymbol{b}$ en étudiant notamment la croissance de la suite $(\deg(V_i))_i$.

Question 5.– Décrire un nouvel algorithme de calcul de polynôme de connexion, et en donner la complexité.

Solutions de l'Exercice 2.

Solution Q1. Voir Algorithme 1.

Solution Q2.

1. On a $B(X) = X + X^2 + X^3 + X^6$.

Algorithme 1 : Algorithme d'Euclide

Entrée : deux polynômes $A, B \in \mathbb{F}[X]$
Sortie : trois polynômes $R, U, V \in \mathbb{F}_q[X]$ tels que $AU + BV = R$ et R est le pgcd de A et B

Initialiser $R_0 \leftarrow A$, $R_1 \leftarrow B$, $U_0 \leftarrow 1$, $U_1 \leftarrow 0$, $V_0 \leftarrow 0$ et $V_1 \leftarrow 1$.
Tant que $R_1 \neq 0$ **faire**
  Calculer le quotient Q de la division euclidienne de R_0 par R_1.
  Mettre à jour $(R_0, R_1) \leftarrow (R_1, R_0 - QR_1)$.
  Mettre à jour $(U_0, U_1) \leftarrow (U_1, U_0 - QU_1)$.
  Mettre à jour $(V_0, V_1) \leftarrow (V_1, V_0 - QV_1)$.
Retourner $R = R_0$, $U = U_1$ et $V = V_1$.

2. Une exécution de l'algorithme de Berlekamp–Massey donne :

k	ℓ_k	$P_k(X)$
0	0	1
1	0	1
2	2	1
3	2	$X+1$
4	2	$X+1$
5	3	X^3+X+1
6	3	X^3+X^2+1
7	3	X^3+X^2+1
8	3	X^3+X^2+1

3. On a

$$\begin{aligned} A(X) &= X^2 \times B(X) + (X^5 + X^4 + X^3) \\ B(X) &= (X+1) \times (X^5 + X^4 + X^3) + (X^2 + X) \\ (X^5 + X^4 + X^3) &= (X^3 + X + 1) \times (X^2 + X) + X \end{aligned}$$

C'est-à-dire :

k	R_k	U_k	V_k
0	X^8	1	0
1	$X + X^2 + X^3 + X^6$	0	1
2	$X^5 + X^4 + X^3$	1	X^2
3	$X^2 + X$	$X+1$	$X^3 + X^2 + 1$
4	X	$X^4 + X^3 + X^2$	$X^6 + X^5 + X^4 + X^3 + X + 1$

4. On observe que pour $k = 3$, on a $\deg R_3 = 2 < 4$, $\deg V_3 = 3 < 4$, et

$$U_3(X)X^8 + V_3(X)B(X) = R_3(X)$$

Autrement dit,

$$V_3(X)B(X) \equiv R_3(X) \mod X^8 \text{ avec } \deg R_3 < 4$$

Le polynôme V_3 est donc un polynôme de connexion de $\boldsymbol{b}$.

Solution Q3. Pour tout i (avant l'arrêt), on a

$$\deg R_{i-1} = \deg Q_i + \deg R_i \quad \text{et} \quad \deg V_{i+1} = \deg Q_i + \deg V_i\,.$$

Donc :

$$\deg V_{i+1} + \deg R_i = \deg V_i + \deg R_{i-1} = \cdots = \deg V_1 + \deg R_0 = 0 + \deg(A)$$

Puis,

$$\deg V_k + \deg R_{k-1} = 2n$$

Solution Q4. La suite $(\deg V_i)_i$ est strictement croissante (avant l'arrêt). On a donc $V_k(X)B(X) \equiv R_k(X) \mod X^{2n}$, avec $\deg R_k < n$ et $\deg V_k \leq 2n - n = n$.

Solution Q5. Si l'on cherche un polynôme de connexion de degré $< n$ d'une suite $\boldsymbol{b}$ dont les $2n$ premiers termes sont $(b_0, \ldots, b_{2n-1})$, on peut :

(i) calculer les polynômes $A(X) = X^{2n}$ et $B(X) = \sum_{i=0}^{2n-1} b_i X^i$,

(ii) exécuter les itérations de l'algorithme d'Euclide étendu, avec comme entrées $A(X)$ et $B(X)$, jusqu'à ce que $R_k(X)$ ait degré $\leq n-1$,
(iii) retourner le coefficient de Bézout $V_k(X)$ associé à $B(X)$.

La complexité de l'algorithme correspond asymptotiquement à celle de l'exécution de l'algorithme d'Euclide (tronquer à $\deg(R_k) \leq n-1$ fait seulement gagner un facteur constant). Elle est donc en $O(n^2)$ opérations dans $\mathbb{F}$.

Remarque. Avec des algorithmes avancés, on peut calculer le pgcd de deux polynômes en temps $O(M(2n)\log n)$, où $M(n)$ représente la complexité de la mutiplication de deux polynômes de degré $\leq n$ sur $\mathbb{F}$.

Exercice 3. ($\star$) Implantation du calcul du polynôme de connexion par l'algorithme d'Euclide.

Question 1.– Implanter l'algorithme d'Euclide étendu, puis l'algorithme de calcul de polynôme de connexion vu dans l'Exercice 2.

Question 2.– Calculer les polynômes de connexion minimaux des suites données dans le fichier `challenges_lfsr.txt` (le même que pour l'exercice du TD1 où vous deviez implanter l'algorithme de Berlekamp–Massey).

Question 3.– Comparer expérimentalement la complexité de votre nouvel algorithme avec celle de l'algorithme de Berlekamp–Massey.

Solutions de l'Exercice 3.

Solution Q1. Voir scripts.

Solution Q2. Voir scripts.

Solution Q3. à venir

Exercice 4. ($\star\star\star$) Implantation de la résolution de système linéaire creux.

Dans cet exercice, on se donne comme objectif de résoudre effectivement un système linéaire creux en temps $O(tn^2)$ et espace $O(nt)$, où la matrice $\boldsymbol{A} \in \mathbb{F}^{n\times n}$ du système a $\leq t$ coefficients non-nuls sur chaque ligne.

Question 1.– Implanter une structure permettant de gérer et d'effectuer des opérations élémentaires sur des matrices creuses : création d'une matrice creuse aléatoire, somme de deux matrices, produit matrice-vecteur, échange de lignes/colonnes, etc.

Question 2.– Implanter une méthode de Horner pour calculer $Q(\boldsymbol{A})\boldsymbol{b}$ en temps $O(ndt)$ et espace $O(nt)$, où $Q(X) \in \mathbb{F}[X]$ est de degré d et $\boldsymbol{b} \in \mathbb{F}^n$.

Question 3.– Implanter une fonction qui calcule le pgcd et le ppcm de deux polynômes de degré $\leq d$ en temps $O(d^2)$.

Question 4.– En s'aidant de l'algorithme de Berlekamp–Massey :

1. implanter une fonction qui calcule le polynôme annulateur d'une suite vectorielle itérée $\boldsymbol{v} = (\boldsymbol{A}^k\boldsymbol{b})_{k\in\mathbb{N}}$;
2. implanter une fonction qui calcule le polynôme annulateur de $\boldsymbol{A}$.

Ces fonctions devront avoir une complexité en $O(tn^2)$ en temps et $O(nt)$ en espace.

Question 5.– Implanter une fonction `get_one_solution(A, b)` qui calcule en temps $O(tn^2)$ et espace $O(nt)$ une solution particulière du système $\boldsymbol{Ax} = \boldsymbol{b}$, où $\boldsymbol{A} \in \mathbb{F}^{n\times n}$ est t–creuse et $\boldsymbol{b} \in \mathbb{F}^n \setminus \{\boldsymbol{0}\}$.

Question 6.– En utilisant l'algorithme de Wiedemann, implanter une fonction `get_kernel_element(A)` qui calcule en temps $O(tn^2)$ et espace $O(nt)$ une solution du système $\boldsymbol{Ax} = \boldsymbol{0}$, où $\boldsymbol{A} \in \mathbb{F}^{n\times n}$ est t–creuse et non-inversible.

Question 7.– Donner les complexités expérimentales (en temps) des fonctions `get_one_solution(A, b)` et `get_kernel_element(A)`. On prendra garde de choisir des valeurs assez grandes de n et assez petites de t (relativement à n) pour observer la croissance en $O(tn^2)$.