Université Paris 8 Année 2022–2023

M2 Mathématiques et applications, parcours ACC

Algorithmes arithmétiques II – Feuille de TD 1

22/09/2022

Le corrigé de certains exercices sera disponible à l'adresse suivante :

`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. $(\star)$ Polynôme de connexion minimal.

Soit $\boldsymbol{u} \in \mathbb{F}^{\mathbb{N}}$ une suite récurrente linéaire sur un corps $\mathbb{F}$.

Question 1.– Démontrer que l'ensemble des polynômes de connexion de $\boldsymbol{u}$ forme un idéal de $\mathbb{F}[X]$.

Question 2.– En déduire qu'il existe un unique polynôme de connexion de $\boldsymbol{u}$ dont le degré est minimal, et tel que $P(0) = 1$.

Exercice 2. $(\star)$ LFSR d'une suite dont les premiers termes sont nuls.

Soit $\boldsymbol{u} \in \mathbb{F}^{\mathbb{N}}$ une suite récurrente linéaire telle que $u_0 = \cdots = u_{k-1} = 0$ et $u_k = 1$. On note $(\ell_n(\boldsymbol{u}))_{n\in\mathbb{N}}$ le profil de complexité linéaire de $\boldsymbol{u}$.

Question 1.– Pour tout $i \in \{1, \ldots, k\}$:

1. démontrer que $\ell_i(\boldsymbol{u}) = 0$;
2. expliciter un polynôme $P_i(X) \in \mathbb{F}[X]$ de degré minimal tel que $(P_i(X), \ell_i(\boldsymbol{u}))$ engendre $\boldsymbol{u}$ sur i termes.

Question 2.– Démontrer que $\ell_{k+1}(\boldsymbol{u}) = k + 1$.

Question 3.– Soit $\boldsymbol{v} \in \mathbb{F}^{\mathbb{N}}$ la suite telle que $v_n = u_{k+n}$ pour tout $n \in \mathbb{N}$. Démontrer que $\ell_n(\boldsymbol{v}) = \ell_{n+k}(\boldsymbol{u}) - k$ pour tout $n \in \mathbb{N}$.

Exercice 3. $(\star\star)$ Exécution de l'algorithme de Berlekamp–Massey.

Question 1.– Dérouler l'algorithme de Berlekamp–Massey sur la suite binaire dont les 10 premiers termes sont :

$$(1, 1, 1, 1, 0, 1, 1, 0, 1, 1)\,.$$

Question 2.– Si la suite se poursuit indéfiniment par la séquence périodique $(0, 1, 1)$, que dire de son polynôme de connexion minimal ?

Exercice 4. $(\star)$ Premiers termes d'une suite définie par sa série formelle.

Soit $\boldsymbol{u} \in \mathbb{F}_2^{\mathbb{N}}$ la suite récurrente linéaire définie par la série formelle

$$U(X) = \frac{1 + X + X^2}{1 + X + X^3}.$$

Question 1.– Quel est l'ordre de la suite ? Combien de termes initiaux possède-t-elle ?

Question 2.– Donner les 15 premiers termes de la suite $\boldsymbol{u}$. Quelle est sa période ?

Exercice 5. 💻 $(\star\star)$ Implantation de l'algorithme de Berlekamp–Massey.

Question 1.– Implanter l'algorithme de Berlekamp-Massey vu en cours, sur un corps fini $\mathbb{F}_2$.

Question 2.– Tester votre fonction avec les 5 suites dont les premiers termes sont donnés dans le fichier `challenges_lfsr.txt`. Les trois premières séquences présentes dans ce fichier sont les suivantes :

1. $(1,1,1,0,1,0,1,1,1,1,0,0,0,1,0,0,1,1,0,1)$
2. $(1,1,0,1,1,1,0,0,1,0,0,0,0,1,1,0,1,1,1,1,0,0,1,0,0,0,0,1,1,0)$
3. $(1,1,1,1,0,1,0,0,0,0,0,1,1,0,0,1,1,1,0,0,1,0,0,0,0,0,1,1,0,0,1,1,1,0,0,1,0,0,0,0,0,...$
   $...1,1,0,0,1,1,1,0,0)$

Question 3.– En produisant des suites linéaires récurrentes aléatoires d'ordre d, donner une estimation expérimentale de la complexité de l'algorithme de Berlekamp–Massey en fonction de d.

Exercice 6. 💻 $(\star\star\star)$ Résolution de systèmes linéaires.

Les fonctions à implanter doivent être suffisamment génériques pour être exécutables sur n'importe quel corps effectif $\mathbb{F}$.

Question 1.– Implanter les fonctions suivantes.

1. Une fonction `right_kernel(T)` qui calcule une base du noyau à droite d'une matrice échelonnée $\boldsymbol{T} \in \mathbb{F}^{n\times n}$. On pourra supposer que les pivots de $\boldsymbol{T}$ sont sur sa diagonale.
2. Une fonction `triangular_solve(T, b)` qui calcule une solution éventuelle d'un système linéaire $\boldsymbol{Tx} = \boldsymbol{b}$, où $\boldsymbol{T} \in \mathbb{F}^{n\times n}$ est sous forme échelonnée. On traitera notamment le cas où le système n'admet aucune solution. Comme pour la question précédente, on pourra supposer que les pivots de $\boldsymbol{T}$ sont sur sa diagonale.
3. Une fonction `gaussian_elimination(A, b)` qui effectue l'élimination gaussienne sur la matrice $\boldsymbol{A} \in \mathbb{F}^{n\times n}$ et le vecteur $\boldsymbol{b} \in \mathbb{F}^n$. Si, dans les questions précédentes, on a supposé que les pivots de $\boldsymbol{T}$ sont sur sa diagonale, alors l'élimination gaussienne devra produire une matrice ayant cette propriété.

Question 2.– Écrire une fonction `solve_system(A, b)` qui calcule l'ensemble des solutions du système d'équations $\boldsymbol{Ax} = \boldsymbol{b}$, où $\boldsymbol{A} \in \mathbb{F}^{n\times n}$ et $\boldsymbol{b} \in \mathbb{F}^n$. On donnera les solutions sous la forme d'un espace affine, dont on décrira un élément particulier et une base de l'espace directeur.

Sur la page web du cours, vous pouvez retrouver des fichiers dont le nom a la forme `system<n>x<n>r<r>q<q>.txt`. Ces fichiers contiennent $n+2$ lignes :

— les n premières lignes sont composés de n chiffres entre 0 et $q-1$, représenant chacun un élément de $\mathbb{F}_q$: chacune de ce lignes représente une ligne de la matrice $\boldsymbol{A}$ du système ;
— la $(n+1)$-ème ligne est vide ;
— la dernière ligne représente $\boldsymbol{b}$ (elle est aussi constituée de n coefficients sur $\mathbb{F}_q$).

La valeur de r donnée dans le nom de fichier représente la dimension de l'espace directeur des solutions au système. Elle est d'ordre indicative, mais peut vous permettre de vérifier la cohérence de vos résultats.

Question 3.– Résoudre les systèmes linéaires donnés dans les fichiers mentionnés ci-dessus.

Question 4.– Implanter une fonction `solve_general_system(A, b)` qui traite le cas où le système n'est pas nécessairement carré, c'est-à-dire lorsque $\boldsymbol{A} \in \mathbb{F}^{m\times n}$ et $\boldsymbol{b} \in \mathbb{F}^m$.

Question 5.– Dans le cas où $\mathbb{F} = \mathbb{F}_2$, donner une estimation numérique la plus précise possible de la complexité de la résolution d'un système linéaire aléatoire de taille $n \times n$. On pourra

— ou bien incorporer des compteurs pour évaluer le nombre d'opérations (additions et multiplications) effectuées en fonction de n,
— ou bien mesurer le temps d'éxecution de l'algorithme.