UNIVERSITÉ DE RENNES 1

Théorie des nombres

TP2 : Algorithme de Berlekamp

Votre TP est à rendre sur l'ENT sous forme d'un fichier PDF contenant le code sage et les résultats correspondants produits par sage en vis a vis. La méthode recommandée est via l'impression d'une feuille de travail du notebook. J'enleverai des points pour des formats de fichiers differents.

Rappel. Méthode pour ouvrir un notebook sage en local :

- taper la commande `sage` dans un terminal
- lorsque sage est lancé, taper `notebook()` (créer un nouveau mot de passe la première fois)
- ensuite ouvrir un navigateur sur la page `http://localhost:8080`
- se connecter sous le login `admin`, et entrer le mot de passe que vous avez créé.

Exercice 1. Préliminaires

1. Définir L comme le corps à 19^3 éléments, avec $a \in L$ tel que $L = F_{17}[a]$. Pour la valeur de a renvoyée par sage, quel est l'ordre de a dans L^* ? a est-il générateur ?
2. Définir l'anneau A des polynômes à coefficients dans L. Soit $Q(X) = X^3 + X^2 - aX + 1$. Quelle est la valeur du coefficient de degré 1500 de $Q(X)^{1000}$?
   Indication : si f est un polynome, $f[10]$ renvoie le coefficient de X^{10}...
3. Comment retrouver l'anneau $L[X]$, l'indéterminée $X \in L[X]$, et le corps L à partir de Q ? *Indication :* utiliser les méthodes `.parent()`, `.gen()` et `.base_ring()`.
4. Déterminer le reste de la division euclidienne de Q^3 par $X^3 + X + a$.
5. Définir l'algèbre MM des matrices 3×3 à coefficients dans L (utiliser `MatrixSpace`). Y inverser la matrice
$$M = \begin{pmatrix} a & 2 & 1 \\ 2 & a & 4 \\ 1 & 9 & 8 \end{pmatrix}$$
puis vérifier que le produit est la matrice identité.

Exercice 2. Algorithme de Berlekamp

Le but est de programmer l'algorithme de Berlekamp pour factoriser un polynôme $P \in k[X]$ sans facteur carré.

1. Definir $k = \mathbb{Z}/29\mathbb{Z}$, $q = \#k$, $R = k[X]$.
2. Etant donné un polynôme $P \in k[X]$, considérons Φ l'application linéaire de $k[X]/(P)$ dans lui même définie par $\Phi : f \mapsto f^q - f$. Écrire une fonction `matrice` qui prend en argument un polynôme $P(X) \in k[X]$ qui renvoie la matrice de Φ dans la base $\{1, X, X^2, \ldots\}$.
   Que s'attend-t-on à trouver pour la 1ere colonne ?

Quelle matrice obtient-on pour le polynôme $P = X^4 + X^3 + X^2 + X + 1$ de $\mathbb{Z}/19\mathbb{Z}[X]$?

3. Ecrire une fonction `basenoyau` qui, étant donné un polynome $P(X)$, détermine une base du noyau de cette matrice, et renvoie la liste L des polynômes correspondant à cette base (*Indication* : `.right_kernel()`).
4. Écrire une fonction `diviseur_mod_cte` qui prend en argument deux polynômes $P(X)$ et $Q(X)$ (on pense que Q dans la liste L ci-dessus), et qui recherche un $s \in k$ tel que le PGCD de $P(X)$ et de $Q(X) - s$ soit non trivial, en renvoyant le diviseur de P ainsi trouvé, et renvoyant `None` s'il n'y en a pas.
5. Programmer une fonction `diviseur` qui prend un polynôme sans facteur carré à coefficients dans un corps fini et qui donne un diviseur non trivial de ce polynôme (éventuellement le polynôme lui-même, ou plutôt *None* (c'est a dire rien du tout), s'il est irréductible).
6. Tester avec $X^4 + 1$ puis avec $X^{12} + X^2 + 1 \in \mathbb{F}_p[X]$ pour les nombres premiers $2 \leqslant p \leqslant 19$. On pourra verifier les resultats avec la methode `.is_irreducible()` ou `.factor()`.

   Que se passe-t-il pour $X^4 + 1$ et $p = 2$?
7. Programmer la fonction `Berlekamp` qui prend un polynôme sans facteur carré à coefficients dans un corps fini, et qui renvoie sa décomposition en produit d'irréductibles. Tester avec $X^{12} + X^2 + 1$ sur F_{19}.

Exercice 3. Séparation par les degrés

1. Programmer un algorithme permettant de factoriser un polynome P à coefficients dans un corps fini en un produit de polynômes sans facteur carré.
2. Lorsque le cardinal du corps de base est grand, l'algorithme de Cantor-Zassenhaus est utile, et il commence par séparer les facteurs du polynôme selon leur degré. Cela se fait par calculs de PGCDs successifs avec les polynômes $X^{q^d} - X$ (où q est le cardinal du corps et $1 \leqslant d \leqslant \frac{\deg P}{2}$).

   Programmer cet algorithme, qui prend un polynome P, et renvoie une liste de polynomes Q_i tel que P est le produit des Q_i, et tel que, pour tout i, tous les facteurs irréductibles de Q_i sont d'un même degré d_i.