Université de Bordeaux — **Calcul formel – Master**
Mathématiques — Année 2022–2023

FEUILLE D'EXERCICES n° 6

Travail sur machine et papier
PGCD, relation de Bézout, théorème des restes chinois

Exercice 1 – [PGCD, BÉZOUT]

1) Calculer à la main pgcd$(312, 793)$ et une relation de Bézout entre ces entiers.

2) Toujours à la main : 15 est-il inversible modulo 38 ? Si oui, calculer son inverse.

3) Essayer les commandes

```
gcd(75,198)
xgcd(75,198)
```

Cette dernière commande donne un triplet (d, u, v). Vérifier que $d = 75u + 198v$. On peut aussi calculer le pgcd de plus de deux entiers en même temps.

```
gcd([75,198,220])
```

Mais si on remplace `gcd` par `xgcd` dans la commande précédente, on n'obtient pas de résultat. Nous écrirons une fonction pour cela dans l'exercice 4.

4) Ces commandes `gcd` et `xgcd` fonctionnent aussi sur les polynômes.

a) Définir l'anneau $\mathbb{Q}[x]$ et calculer pgcd$(x^7 - 1, 2x^3 + 5x^2 - 7)$, ainsi que les coefficients de Bézout associés. Vérifier la relation de Bézout correspondante.

b) Même exercice avec pgcd$(x^7 - 1, x^3 - x^2 + 1)$. La classe de $x^3 - x^2 + 1$ dans $\mathbb{Q}[x]/(x^7 - 1)$ est-elle inversible ? Si tel est le cas, quelle est son inverse ?

c) Même exercice dans $\mathbb{F}_7[x]$ avec pgcd$(x^7 - x, x^4 + x^2 - 2)$, puis pgcd$(x^7 - x, x^4 + x^2 + 1)$.

Exercice 2 – [RESTES CHINOIS]

On considère les systèmes

$$(1)\ \begin{cases} 12x - 12 \equiv 6 \bmod 33 \\ 7x + 6 \equiv 7 \bmod 13 \\ 6x - 21 \equiv 9 \bmod 54 \end{cases} \qquad (2)\ \begin{cases} 15x \equiv 7 \bmod 25 \\ 8x \equiv 1 \bmod 13 \\ 7x \equiv 4 \bmod 11 \end{cases}$$

$$(3)\ \begin{cases} x \equiv 5 \bmod 21 \\ x \equiv 3 \bmod 28 \\ x \equiv 1 \bmod 5 \end{cases} \qquad (4)\ \begin{cases} x \equiv 3 \bmod 21 \\ x \equiv 17 \bmod 49 \\ x \equiv 1 \bmod 5 \end{cases} \qquad (5)\ \begin{cases} 3x + 1 \equiv 0 \bmod 5 \\ 4x + 2 \equiv 1 \bmod 7 \\ x - 1 \equiv 1 \bmod 4 \end{cases}$$

1) Écrire le système (1) sous forme de problème des restes chinois et le résoudre en utilisant la formule du cours.

2) Pourquoi les systèmes (2) et (3) n'ont-ils pas de solutions ?

3) Dans le système (4), les moduli ne sont pas deux à deux premiers entre eux. Donner un système équivalents où les moduli sont deux à deux premiers entre eux et le résoudre.

4) Retrouver les résultats précédents en appliquant la commande `crt` aux systèmes (1) (mis sous forme de problème des restes chinois), (3) et (4).

Note. Cette commande `crt` s'applique à tout anneau euclidien par exemple à $k[x]$, où k est un corps.

Note. On connaît bien l'algorithme correspondant dans le cas où les moduli sont deux à deux premiers entre eux. L'exercice 6 de la feuille 6 montre comment on peut adapter l'algorithme au cas de moduli non deux à deux premiers entre eux.

Exercice 3 – [Restes chinois et interpolation]

Dans cet exercice, la question 1) sera faite d'abord sur papier. Ensuite, toutes les questions seront faites sur machine en utilisant la commande "crt" (après un premier travail de traduction des contraintes en termes de congruences).

1) Déterminer le polynôme P de $\mathbb{Q}[x]$ de degré inférieur ou égal à 2 tel que

$$P(0) = 2,\ P(1) = 2,\ P(2) = 1.$$

2) Déterminer le polynôme P de $\mathbb{Q}[x]$ de degré inférieur ou égal à 3 tel que

$$P(0) = 0,\ P'(0) = 1,\ P(1) = 1,\ P'(1) = 0.$$

3) Déterminer le polynôme P de $\mathbb{Q}[x]$ de degré inférieur ou égal à 4 tel que

$$P(0) = -1,\ P(1) = 1,\ P(2) = 7,\ P'(1) = 3,\ P''(1) = 1.$$

4) Déterminer le polynôme P de $\mathbb{F}_7[x]$ de degré inférieur ou égal à 4 tel que

$$P(0) = 2,\ P(1) = 2,\ P(2) = -1,\ P(-1) = 1,\ P'(1) = 0.$$

Exercice 4 – [Relation de Bézout pour plus de deux entiers] Comme on l'a vu ci-dessus, le logiciel sage permet de calculer le pgcd de deux entiers à l'aide de la commande `gcd`. La commande `xgcd` donne en plus les coefficients de Bézout. On a vu aussi que la commande `gcd([a1,...,an])` donne $\mathrm{pgcd}(a_1, \ldots, a_n)$, mais que par contre, `xgcd([a1,...,an])` ne donne rien.

En utilisant `xgcd`, écrire un algorithme qui prend en entrée une famille d'entiers $(a_1, \ldots, a_n)$ et donne en sortie $d = \mathrm{pgcd}(a_1, \ldots, a_n)$ et $(u_1, \ldots, u_n) \in \mathbb{Z}^n$ tel que

$$\sum_{i=1}^{n} a_i u_i = d.$$

Exercice 5 – [Complexité de l'Algorithme d'Euclide étendu pour les entiers]

On rappelle l'algorithme d'Euclide étendu.

Algorithme 1. Algorithme d'Euclide étendu pour les entiers

Entrées: a, $b \in \mathbb{N}$, $(a,b) \neq (0,0)$
Sorties: $\text{pgcd}(a,b)$ et u, $v \in \mathbb{Z}$ tels que $au + bv = \text{pgcd}(a,b)$

$u_0 = 1$, $v_0 = 0$, $r_0 = a$
$u_1 = 0$, $v_1 = 1$, $r_1 = b$
$i = 1$ $\quad \{initialisations\}$
tantque $r_i \neq 0$ **faire**
  Division de r_{i-1} par $r_i \to$ quotient q_i et reste r_{i+1}
  $u_{i+1} = u_{i-1} - q_i u_i$
  $v_{i+1} = v_{i-1} - q_i v_i$
  $i = i + 1$
Retourner le dernier r_i non nul ainsi que les u_i et v_i correspondants

1) Soit $U_i = \begin{pmatrix} u_i & v_i \\ u_{i+1} & v_{i+1} \end{pmatrix}$. Vérifier que pour tout i,

$$U_i = \begin{pmatrix} 0 & 1 \\ 1 & -q_i \end{pmatrix} \cdots \begin{pmatrix} 0 & 1 \\ 1 & -q_1 \end{pmatrix} \quad \text{et} \quad \begin{pmatrix} r_i \\ r_{i+1} \end{pmatrix} = U_i \begin{pmatrix} a \\ b \end{pmatrix}$$

2) Montrer que $U_i^{-1} = \begin{pmatrix} |v_{i+1}| & |v_i| \\ |u_{i+1}| & |u_i| \end{pmatrix}$.

3) En déduire que pour tout $i > 0$, $|u_i| \leqslant b/r_{i-1}$ et $|v_i| \leqslant a/r_{i-1}$.

4) Montrer que la complexité binaire de cet algorithme est quadratique.

Exercice 6 – [Inversion rapide modulo x^n] Soit K un corps. Soient $F \in K[x]$ et n un entier naturel non nul. On suppose que $F(0) = 1$. On sait que dans ce cas, la classe de F dans $K[x]/(x^n)$ est inversible. Cet exercice porte sur un algorithme rapide pour calculer son inverse.

Soit (A_i) la suite définie de la manière suivante.

$$A_0 = 1 \quad , \quad A_{i+1} = 2A_i - FA_i^2 \quad \forall i \geqslant 0.$$

1) Montrer que $FA_i \equiv 1 \mod x^{2^i}$. pour tout $i \geqslant 0$.

2) En déduire un algorithme `Inverse(n,F)` pour calculer l'inverse de F modulo x^n. On s'efforcera d'optimiser la complexité de cette fonction.

3) On suppose que la complexité algébrique de la multiplication de deux polynômes de $K[x]$ de degrés inférieurs à un entier N est en $O(N \log N)$. On rappelle que prendre le reste de la division d'un polynôme par x^N revient à tronquer le polynôme : on ne prend pas cette opération en compte dans le calcul de la complexité algébrique.

Soient $C(n)$ la complexité algébrique de `Inverse(n,F)` et $r = \lceil \log n \rceil$ le plus petit entier supérieur ou égal à $\log n$. Montrer que $C(n)$ est en $O(r2^r)$. En déduire que $C(n)$ est en $O(n \log n)$.

Exercice 7 – [Division euclidienne rapide]
Soit K un corps. pour tout polynôme $F \in K[x]$, et tout entier k, on note

$$\mathrm{rev}_k(F)(x) = x^k F\left(\frac{1}{x}\right).$$

1) Montrer que si $k \geqslant \deg F$, alors $\mathrm{rev}_k(F) \in K[x]$ et $\mathrm{rev}_k(\mathrm{rev}_k(F)) = F$.

2) Soient F et G deux polynômes non nuls de $K[x]$ de degrés respectifs m et n tels que $m \geqslant n$. Soient Q et R le quotient et le reste de la division euclidienne de F par G. Montrer que

$$\mathrm{rev}_m(F) = \mathrm{rev}_{m-n}(Q)\mathrm{rev}_n(G) + x^{m-n+1}\mathrm{rev}_{n-1}(R).$$

3) Montrer que la classe de $\mathrm{rev}_n(G)$ dans $K[x]/(x^{m-n+1})$ est inversible.

4) On suppose que les multiplications de polynômes de $K[x]$ de degrés inférieurs à N peuvent se faire en $O(N \log N)$. Déduire de l'exercice 6 un algorithme de complexité quasi-linéaire pour la division euclidienne dans $K[x]$.