# 1 À retenir

- Complexité pratique des opérations élémentaires dans $\mathbb{Z}$ et $K[x]$,
- Définition du pgcd, du ppcm, théorème de Bezout,
- Connaître parfaitement et *sans note* l'algorithme d'Euclide étendu, pour les entiers et les polynômes,
- Savoir analyser un algorithme élémentaire, par exemple un algorithme de tri, ou un algorithme simple pour les graphes,
- Algorithmique élémentaire de $(\mathbb{Z}/N\mathbb{Z})$.

# 2 Étude expérimentale de quelques complexités

En utilisant le compteur de PARI/GP étudier la complexité (temps de calcul en fonction de la taille des données) pour les opérations suivantes

1. addition de deux entiers de $n$ chiffres décimaux,
2. multiplication de deux entiers de $n$ chiffres décimaux,
3. calcul de $a^b \bmod c$ où $a$, $b$ et $c$ sont trois entiers de $n$ chiffres décimaux,
4. multiplication de deux matrices carrées $d \times d$ à coefficients dans $\mathbb{Z}/n\mathbb{Z}$.

Si l'on devine une fonction de complexité, par exemple du type

$$T(n) = u + vn^{\alpha},$$

on cherchera à déterminer les trois constantes $u$, $v$ et $w$ aussi précisément que possible.

On pourra ensuite étudier la complexité de certaines fonctions de PARI/GP telles que **isprime**, ou encore **factor**, ou encore **nextprime**, ou encore **vecsort**.

# 3 Algorithmes de tri, algorithme de Shanks

Programmer un algorithme lent et un algorithme rapide pour trier une liste d'entiers. Étudier et comparer leurs complexités pratiques.

Programmer l'algorithme de Shanks pour le calcul du logarithme discret. Pour la recherche de collision, on utilisera successivement la fonction vecsort de PARI/GP et les deux algorithmes implémentés ci-dessus. Conclusion ?

En vous appuyant sur les estimations obtenues, donner une idée de la taille de clé nécessaire pour un cryptosystème basé sur le problème du logarithme discret dans un groupe où il n'existe pas d'algorithme plus rapide que celui de Shanks.

# 4 Le groupe $(\mathbb{Z}/N\mathbb{Z})^*$

Programmer en PARI/GP une procédure qui calcule un générateur de $(\mathbb{Z}/N\mathbb{Z})^*$ pour un entier premier $N$ donné. Quelle est la complexité de cette procédure ? Où réside la difficulté ?

Donner une procédure qui fabrique (resp. qui vérifie) un certificat de primalité de type Pocklington-Lehmer. Quelles sont les complexités de ces procédures ?

# 5 Graphes

On se donne un graphe non orienté $(E, V)$ avec $V \subset E \times E$.
Programmer un algorithme pour calculer la composante connexe d'un sommet.
Programmer un algorithme pour calculer une base des cycles.