TP 10 – TRANSFORMÉE DE FOURIER RAPIDE (BONUS)

► **PRÉREQUIS** : Chapitre 9 du poly.

1 Exercice I

Soit $n = 2^k$ une puissance de 2 avec $k > 0$. Soit ω une racine primitive n -ième de l'unité. On définit un isomorphisme de $\mathbb{C}$ -algèbres $\mathcal{F}_\omega : \mathbb{C}[X]/(X^n - 1) \rightarrow \mathbb{C}^n$ par

$$\mathcal{F}_\omega(R) = (R(1), R(\omega), \dots, R(\omega^{n-1}))$$

où l'on identifie la classe $\bar{R} \in \mathbb{C}[X]/(X^n - 1)$ avec un représentant $R \in \mathbb{C}[X]$.

On évalue $\mathcal{F}_\omega(R)$ en calculant récursivement l'image de deux polynômes de degré $< m = \frac{n}{2}$ via les formules

$$R(\omega^p) = \sum_{j=0}^{m-1} R_{2j} \alpha^{jp} + \omega^p \sum_{j=0}^{m-1} R_{2j+1} \alpha^{jp} \quad \text{et} \quad R(\omega^{p+m}) = \sum_{j=0}^{m-1} R_{2j} \alpha^{jp} - \omega^p \sum_{j=0}^{m-1} R_{2j+1} \alpha^{jp}$$

où l'on identifie un polynôme R de degré $< n$ avec la liste de ses coefficients $(R_0, \dots, R_{n-1})$ et où $0 \leq p < m$ et $\alpha = \omega^2$.

Programmer l'algorithme récursif s'appuyant sur cette remarque. La procédure FFT recevra en entrée R, ω et n et retournera $\mathcal{F}_\omega(R)$.

2 Exercice II

On rappelle qu'avec les notations précédentes, si P et Q sont deux polynômes de $\mathbb{C}[X]$ tels que $\deg(PQ) < n$ alors

$$\mathcal{F}_\omega(PQ) = \mathcal{F}_\omega(P)\mathcal{F}_\omega(Q)$$

et que pour tout polynôme R de degré $< n$, on a

$$\mathcal{F}_{\omega^{-1}}(\mathcal{F}_\omega(R)) = \mathcal{F}_\omega(\mathcal{F}_{\omega^{-1}}(R)) = nR.$$

Écrire une procédure qui prend en entrée P, Q et n et renvoie PQ en utilisant votre fonction FFT de la question précédente.

3 Exercice III

Expérimenter les commandes suivantes

```
A=[RR(1) for i in range(8)]
s=IndexedSequence(A, range(8))
t=s.fft();t
lt=t.list();lt
```

Comparer avec les résultats de votre exercice II.

4 Exercice IV

Appliquer l'algorithme des exercices II et III à un corps fini, par exemple $\mathbb{F}_{257}$ et $n = 16$.