TUTORIAL 10 – REVISION

1 Time to read your lessons

1. Fill the following table without using the fast algorithms.

	$\mathbb{Z}$	$K[X]$
What we count in the complexity		
Size of the input	a is an integer	A is a polynomial
Addition/Subtraction		
Multiplication		
Euclidean division		
Extended GCD		
Multiple point evaluation		
Interpolation/CRT		

2. We denote by $M(n)$ the complexity of multiplying two integers (resp. two polynomials) of size n (resp. of degree n). What is the best value you know for $M(n)$ (beware, this is not the same for polynomials and integers)?

3. Fill the following table using the fast algorithms.

	$\mathbb{Z}$	$K[X]$
Addition/Subtraction		
Multiplication		
Euclidean division		
Extended GCD		
Multiple point evaluation		
Interpolation/CRT		

4. What is the complexity of adding/multiplying polynomials in $\mathbb{F}_q[X]$ in terms of bit operations (use naive algorithms).

5. Fill the following table for linear algebra.

	$M_n(K)$	$M_n(K)$ sparse (ω non zero coefficients)
What we count in the complexity		
Size of the input		
Addition/Subtraction		
Matrix $\times$ vector		
Multiplication		
Gauss Pivoting		
Determinant		
Linear systems		
Inversion		
Characteristic polynomial		

2 Evaluating the derivatives of a polynomial at some point

In this exercise, we are given a degree n polynomial $P \in K[X]$ for some field K and a point a in K. Our objective is to evaluate all the derivatives of P at point a.

1. Give a naive algorithm that computes $P^{(i)}(a)$ for all $i \in \{0, \cdots, n\}$. What is its complexity ?

2. If $a = 0$, give an algorithm that computes $P^{(i)}(0)$ for $0 \leq i \leq n$ in linear time.

3. Recall an algorithm that computes $Q(X) = P(X + a)$ from P in quasi-linear time. (Hint: you may want to compute the Euclidean division $P(X) = u(X)(X - a)^{\deg(P)/2} + v(x)$ and apply a recursive algorithm).

4. Conclude by giving a quasi-linear time algorithm that computes $P^{(i)}(a)$ for all $i \in \{0, \cdots, n\}$.

3 Hensel-type strategy for solving linear system

In this exercise, we study algorithms to solve $Mx = b$, $M \in \mathcal{M}_n(K[X]), b \in K[X]^n$. We shall assume that the degree of all coordinates of M, b is $\leqslant d$.

Cramer's formulas show that if x is a solution of $Mx = b$, $(\det M) \cdot x \in K[X]^n$, and the coefficients of $(\det M) \cdot x$ have degree $\leqslant nd$. We'll also assume that $\det M(u) \neq 0$ for all $u \in K$.

1. What is the complexity of computing $B := (M \bmod X)^{-1}$?

   Let $y_i \in K[X]^n$ be a solution of $My_i = b \bmod X^i$, and define $r_i = b - My_i$.

2. Prove that $r_i = \lambda_i X^i$ for some $\lambda_i \in K[X]^n$. If $z_i = B\lambda_i \bmod X$, prove that $y_{i+1} = y_i + X^i z_i$ and $r_{i+1} = r_i - X^i M z_i$.

3. What is the complexity of computing y_{nd+1} using this method? Assuming that $\det M$ is given as input or precomputed, deduce an algorithm for solving $Mx = b$.

4. If we need to compute $\det M$ beforehand, then this computation is going to dominate the complexity of linear system solving. Can we avoid computing the determinant? (Hint: use rational reconstruction.)

4 One question / one minute

1. Find the smallest $u \geq 0$ such that
$$\begin{cases} u \equiv 1 \mod 2 \\ u \equiv 2 \mod 3 \\ u \equiv 2 \mod 5 \end{cases}$$

2. Explain why you can see Karatsuba's algorithm as an evaluation/interpolation algorithm.

3. Let K be a field, $a_1, \cdots, a_n$ be elements of K and $(p_0, \cdots, p_{n-1}) \in K^n$. We want to compute the matrix-vector product
$$\begin{pmatrix} 1 & a_1 & a_1^n & \cdots & a_1^{n-1} \\ 1 & a_2 & a_2^n & \cdots & a_2^{n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & a_n & a_n^n & \cdots & a_n^{n-1} \end{pmatrix} \begin{pmatrix} p_0 \\ p_1 \\ \vdots \\ p_{n-1} \end{pmatrix}.$$
   What is the best complexity you can achieve to compute this matrix-vector product? What if ω is a n-th root of unity with n a power of 2 and $a_i = \omega^i$?