TUTORIAL 10

1 Time to read your lecture notes

1. Let K be a field. Fill the following table **without** using fast algorithms.

	$\mathbb{Z}$	$K[X]$
unit operation in complexity		
input size		
Addition/subtraction		
Multiplication		
Euclidean division		
Extended Euclidean Algorithm		
Multipoints evaluations		
Interpolation/CRT		

2. Let $\mathrm{M}(n)$ be the complexity of multiplying two integers of size n, and $\mathrm{P}(n)$ be the complexity of multiplying two polynomials of degree n. What are the best values you know for each?

3. What is the complexity of adding/multiplying polynomials in $\mathbb{Z}_p[X]$ in terms of bit operations (and using naive algorithms)?

4. Same as question 1, but fast algorithms are allowed this time.

	$\mathbb{Z}$	$K[X]$
unit operation in complexity		
input size		
Addition/subtraction		
Multiplication		
Euclidean division		
Extended Euclidean Algorithm		
Multipoints evaluations		
Interpolation/CRT		

5. Fill the following table for linear algebra.

	$\mathcal{M}_n(K)$	sparse (say, c non-zero coeffs)
unit operation in complexity		
input size		
Addition/subtraction		
Matrix $\times$ vector		
Matrix multiplication		
Gaussian elimination		
Determinant		
Linear systems		
Inversion		
Characteristic polynomial		

2 One-question, one-minute

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

2. Explain how Karatsuba's algorithm can be seen as an evaluation/interpolation algorithm.

3. Let K be a field, $c = (c_0, \dots, c_{n-1}) \in K^n$ and for $a_1, \dots, a_n$ elements in K, consider the matrix-vector product
$$\begin{bmatrix} 1 & a_1 & \dots & a_1^{n-1} \\ 1 & a_2 & \dots & a_2^{n-1} \\ & \vdots & & \\ 1 & a_n & \dots & a_n^{n-1} \end{bmatrix} \cdot \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{bmatrix}.$$
What is the best complexity you can achieve? Can you do even better if you ask an additional property on the a_i's?

3 Newton's iteration and integer roots

Newton's iteration is a fondamental tool in several topics (optimal transport, numerical analysis, ...). It can be seen as an iterative process to approximate a solution of $f(x) = 0$, where f is, *e.g.* , a given $\mathcal{C}^1$- real valued function. Starting from an initial condition x_0, one defines

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}.$$

Under some reasonable assumptions on f, it can be shown that the process converges quadratically fast to a solution. With a bit more work, one can show that Euclidean division of two integers of bitsize n can be done in $O(\mathrm{M}(n))$: we will use this result in the exercise.

We will also show that Newton's iteration can also be used with another notion of convergence to compute integer roots of integer polynomials.

1. Write down Newton's iteration to compute the inverse of a given $a \in \mathbb{Z}$. Use it to show that, if you start from an inverse of a modulo some integer m, the next iteration gives you an inverse of a modulo m^2. Deduce an algorithm to compute inverses modulo p^v, for some prime p and an even[1] integer $v \geq 1$, and give its complexity.

2. Let $P \in \mathbb{Z}[X]$, and assume that $a_i \in \mathbb{Z}$ such that $P(a_i) \equiv 0\ [m]$ and $P'(a_i) \in (\mathbb{Z}/m\mathbb{Z})^\times$ for some integer m. Define the next "modular" Newton iterate a_{i+1}, and show that it satisfies the following properties:

   - $a_{i+1} \equiv a_i\ [m]$;
   - $P(a_{i+1}) \equiv 0\ [m^2]$;
   - $P'(a_{i+1}) \in (\mathbb{Z}/m^2\mathbb{Z})^\times$.

   The next question deals with the unicity of the solution given by the above Newton iteration process, assuming b_0 is a starting value (thus $P(b_0) \equiv 0\ [p]$ and $P'(b_0)$ is invertible modulo p).

3. (**Uniqueness property**) If b, b' are integers such that $b \equiv b_0 \equiv b'\ [p]$ and $P(b) \equiv P(b')\ [p^{2^i}]$, show that $b \equiv b'\ [p^{2^i}]$.

4. What is missing to design a general algorithm to compute "modular" roots of integer polynomials? Assuming this problem can be dealt with, write down an algorithm that compute modular roots of integer polynomials and give its complexity.

4 Evaluating the derivatives of a polynomial

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

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

2. If $a = 0$, give an algorithm that computes all $P^{(i)}(0)$'s in linear time.

3. Show that $Q(X) = P(X + a)$ can be computed in quasi-linear time. (Hint: Consider the Euclidean division $P(X) = Q(X)(X-a)^{\deg P/2} + R(X)$, and apply a recursive approach.)

4. Conclude with a quasi-linear time algorithm to compute all $P^{(i)}(a)$'s.

[1] For the sake of simplicity. An analogous algorithm can be designed in the general case (we leave it as an exercise).