TUTORIAL 5

In all the exercises, $\mathbb{K}$ be a commutative field of characteristic not equal to 2 (the FFT is quite tricky to work out in characteristic 2 – no nice roots of unity), we shall assume that all operations in K cost $O(1)$, and $M(n)$ stands for the complexity of multiplying two polynomials of degree n.

1 FFT as a particular multipoint evaluation

1. Let $n = 2^k \in \mathbb{N}$, and P and Q be two polynomials of $K[X]$ with degree at most $n/2 - 1$. Explain why the FFT algorithm for multiplying P and Q is a particular case of the fast multipoint evaluation algorithm.
2. Recall what is the complexity of multiplying P and Q using the FFT algorithm. What is the general complexity of fast multi-point evaluation at n points? Why is the complexity of the FFT algorithm better than in the general fast multipoint evaluation algorithm?

2 Fast CRT

1. Recall (any version of) the Chinese Remainder Theorem.

   Let $P_i \in K[X]$ for $i \in \{0, \ldots, k-1\}$ be pairwise coprime polynomials, with $d_i := \deg P_i$. Let $N = \prod_{i=0}^{k-1} P_i$ and $n := \sum_{i=0}^{k-1} d_i = \deg N$.

   Note some useful properties of $M(n)$: $\sum_{i=0}^{k-1} M(d_i) \leq M(n)$ (M is superlinear) and $M(2n) = O(M(n))$.
2. Let $u_0, \ldots, u_{k-1}$ be polynomials with $\deg u_i < d_i$. Give an algorithm of complexity $O(M(n) \log n \log k)$ to compute a polynomial x of degree $< n$ such that
$$x = u_i \mod P_i \ \forall i \in [k]. \tag{1}$$
   (Bonus: Note that your algorithm works in the integer case (if P_i and u_i are integers).)
3. Prove that one can compute all the polynomials $R_i := N \bmod P_i^2$ in time $O(M(n) \log k)$ (generalize fast multipoint evaluation).
4. Define $S_i = (R_i/P_i)^{-1} \bmod P_i$. Show that S_i is well defined (i.e. R_i/P_i is invertible modulo P_i) and that one can compute all the S_i's in time $O(M(n) \log n)$.
5. Prove that $x = \sum_{i=0}^{k-1} c_i N/P_i$ with $c_i = u_i S_i \bmod P_i$ is a solution to question 2, and explain how to compute x in time $O(M(n) \log n)$ – try to use a similar strategy to the one that was used during the class for evaluating Lagrange's formula in quasilinear time.

3 Determinant

Let $M \in \mathcal{M}_n(\mathbb{K}[X])$. Assume that all the entries of M have degree at most d. Give an evaluation interpolation algorithm for computing $\det(M)$. What is its complexity ?

4 Hermite Interpolation

For $i \in \{1, \cdots, n\}$, let $(x_i, y_i, z_i) \in \mathbb{K}^3$ with x_i pairwise distinct. An Hermite interpolating polynomial for (x_i, y_i, z_i) is a polynomial P of degree $\leq 2n - 1$ such that $P(x_i) = y_i$ and $P'(x_i) = z_i$.

1. Show that such a P exists and is unique.
2. Give an algorithm to find P. What is the complexity of this algorithm? Hint: Try to generalize Newton's algorithm for interpolation. (You should not give the same algorithm as in next question).
3. Use Exercise 2 to give a quasi-linear time algorithm (Hint: try to express the constraints $P(x_i) = y_i$ and $P'(x_i) = z_i$ as a unique constraint of the form $P \equiv Q_i \mod (X - x_i)^2$ for some polynomial Q_i of degree 1).
4. Can you state a generalization to higher order derivatives? With a different order at each point?