TUTORIAL 7

1 Déjà vu

In this exercise, $x_1, x_2, \dots, x_n$ are elements of K and P, Q are polynomials in $K[X]$ of degree $< n$.

1. Let $h(X) = \prod_{j=1}^{n}(X - x_j)$. Give a quasi-linear algorithm for computing $P \circ Q \bmod h$.
2. When h is an arbitrary polynomial of degree n, what is the best complexity you can achieve for computing $P \circ Q \bmod h$?

2 Fast characteristic polynomial

Let A be an $n \times n$ matrix. In this exercise, we will denote by n^ω the number of operations in K needed to multiply two n by n matrices with coefficients in K. You will see in class that given a n by n matrix $M \in \mathcal{M}_n(K)$, we can compute M^{-1} using $O(n^\omega)$ operations in K (computing the inverse is asymptotically the same as multiplying).

1. Assume that v is a vector such that v, $Av, A^2v, \dots, A^{n-1}v$ is a basis of K^n; then if B is the matrix with columns $v, Av, A^2v, \dots, A^{n-1}v$, prove that $B^{-1}AB$ is a *companion matrix*, that is, a matrix of the following form.

$$C = \begin{bmatrix} 0 & 0 & \cdots & 0 & c_0 \\ 1 & 0 & \cdots & 0 & c_1 \\ 0 & 1 & \cdots & 0 & c_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & c_{n-1} \end{bmatrix}.$$

2. If B is given, what is the cost of computing the characteristic polynomial of A using the previous question.
3. Explain why from an $n \times n$ matrix multiplication in time $O(n^\omega)$ we can deduce a $n \times m$ by $m \times k$ matrix multiplication algorithm in time $O(\max(n, m, k)^\omega)$
4. Define $w_0 = v, w_1 = (v, Av), w_2 = (v, Av, A^2v, A^3v), \dots, w_k = (v, Av, A^2v, \dots, A^{2^k-1}v)$

   Prove that w_k can be computed in time $O(kn^\omega)$ for $k < \log n$.
5. Under the assumption that v exists and that you know it, give a $O(n^\omega \log n)$ algorithm for computing the characteristic polynomial of a square matrix.
6. Does there always exist a v as in question 1 ?

Remark. A good final (but purely mathematical) question is to show that on the other hand, if the characteristic polynomial of A is irreducible, any nonzero v works.

3 Sylvester matrices

Let K be a field, and $P = \sum_{i=0}^{d_P} p_i X^i$, $Q = \sum_{i=0}^{d_Q} q_i X^i$ be two polynomials in $K[X]$ of respective degree d_P and d_Q. Put $D = d_P + d_Q$, define $v_P = (p_0, p_1, \ldots, p_{d_P}, 0, \ldots, 0) \in K^D$ and $v_Q = (q_0, q_1, \ldots, q_{d_Q}, 0, \ldots, 0) \in K^D$.

For $x = (x_0, \ldots, x_{D-1})$ a vector in K^D, define $C(x) = (0, x_0, \ldots, x_{D-2})$. The *Sylvester matrix* of P and Q is the matrix of size D whose colums are

$$(v_P, C(v_P), \ldots, C^{d_Q-1}(v_P), v_Q, C(v_Q), \ldots, C^{d_P-1}(v_Q)).$$

It is probably better illustrated on an example: if P has degree 2 and Q degree 3, then we have

$$S(P,Q) := \begin{pmatrix} p_0 & 0 & 0 & q_0 & 0 \\ p_1 & p_0 & 0 & q_1 & q_0 \\ p_2 & p_1 & p_0 & q_2 & q_1 \\ 0 & p_2 & p_1 & q_3 & q_2 \\ 0 & 0 & p_2 & 0 & q_3 \end{pmatrix}.$$

1. Let $v = (v_0, \ldots, v_{d_Q-1}, w_0, \ldots, w_{d_P-1}) \in K^D$. Compute $S(P,Q) \cdot v$ and express it in terms of the polynomials $V = \sum v_i X^i$ and $W = \sum w_i X^i$.

2. What is the best complexity you can achieve for computing a product $S(P,Q) \cdot v$ using fast arithmetic?

3. If P, Q are coprime, what is the best complexity you can achieve for solving the equation $S(P,Q) \cdot v = w$? Or computing the inverse of $S(P,Q)$?