Homework 2

Date: Aug 23, 2021 — Algebra & Computation-F21
Instructor: Mrinal Kumar — Due: 5 pm, Sep 13, 2021

Instructions

- It is slightly preferred that you type your homeworks up in LaTeX. In case you turn in scans of handwritten notes, please make sure that they are legible.
- Discussion on the problems with other members of the class is permitted and to an extent, even encouraged. But, you *must* write the solutions on your own. You *must* also acknowledge any discussions you might have had with others separately for every problem.
- Please do not look up solutions on the internet or in other references. In case you use any sources outside the notes for this course, again properly acknowledge them.
- To get the most out of the problem sets, you are encouraged to think about the problems on your own before discussing them with others, consulting the references or looking at the hints (which some of the problems might have).

Problems

1. Let q be a prime power and let $k > 0$ be a natural number. The polynomial $\mathsf{Trace}(x)$ is defined as
$$\mathsf{Trace}(x) = x + x^q + x^{q^2} + \cdots + x^{q^{k-1}} .$$
   (a) **(5 points)** Show that for every $\alpha \in \mathbb{F}_{q^k}$, $\mathsf{Trace}(\alpha) \in \mathbb{F}_q$.
   (b) **(5 points)** Show that when viewed as a map from the vector space $\mathbb{F}_{q^k}$ to $\mathbb{F}_q$, Trace is $\mathbb{F}_q$ linear. [1]
   (c) **(10 points)** Using the properties of Trace, conclude that for *every* $\mathbb{F}_q$ linear map L from $\mathbb{F}_{q^k}$ to $\mathbb{F}_q$, there is an $\alpha_L \in \mathbb{F}_{q^k}$ such that for all $\beta \in \mathbb{F}_{q^k}$, $L(\beta) = \mathsf{Trace}(\alpha_L \cdot \beta)$.[2]

2. Let $\mathbb{F}$ be a field. We saw in the first homework that for any set of points $(\alpha_1, \beta_1), \ldots, (\alpha_n, \beta_n)$ in $\mathbb{F} \times \mathbb{F}$ with $\alpha_i \neq \alpha_j$ for all $i \neq j$, there is a unique polynomial $f \in \mathbb{F}[x]$ of degree at most $n-1$ such that for every $i \in \{1, 2, \ldots, n\}$, $f(\alpha_i) = \beta_i$. Moreover, as many of you noted in your solutions, this polynomial is equal to
$$P(x) = \sum_{i=1}^{n} \beta_i \cdot \prod_{j \in \{1,2,\ldots,n\} \setminus \{i\}} \frac{(x - \alpha_j)}{(\alpha_i - \alpha_j)} .$$
   For this problem, the goal is to design an algorithm that runs in time $O(n \cdot \mathsf{poly}(\log n))$, and on input $\{(\alpha_i, \beta_i) : i \in \{1, 2, \ldots, n\}\}$ as above, outputs the polynomial P as a list of coefficients.
   For this, it might be helpful to note that $P(x)$ can be rewritten as
$$P(x) = \sum_{i=1}^{n} \beta_i \cdot u_i \cdot M(x)/(x - \alpha_i) ,$$

[1] Recall that sometimes we view the field $\mathbb{F}_{q^k}$ as the k dimensional vector space over $\mathbb{F}_q$.
[2] The multiplication in $\alpha_L \cdot \beta$ is being done over the field $\mathbb{F}_{q^k}$.

where, $u_i = \prod_{j \in \{1,2,\ldots,n\} \setminus \{i\}} \frac{1}{(\alpha_i - \alpha_j)}$, and $M(x)$ is the polynomial $\prod_{j=1}^{n}(x - \alpha_j)$.

(a) **(10 points)** As a first step towards a fast algorithm for polynomial interpolation, design an algorithm that outputs $u_1, u_2, \ldots, u_n$ and runs in time $O(n \cdot \mathsf{poly}(\log n))$.

(b) **(10 points)** Use the algorithm in the previous part along with other ideas from the class to design an $O(n \cdot \mathsf{poly}(\log n))$ time algorithm for univariate polynomial interpolation.

3. **(10 points)** This question is again based on Question 3 in the first homework. We saw there for every $n \in \mathbb{N}$ and for every set of points $\{(\alpha_i, \beta_i) : i \in \{1, 2, \ldots, n\}\} \subseteq \mathbb{F}^2$ with $\alpha_i \neq \alpha_j$ for all $i \neq j$, there is a unique univariate polynomial P of degree at most $n-1$ such that for all $i \in \{1, 2, \ldots, n\}$, $P(\alpha_i) = \beta_i$. We also noticed in the class that this problem can be viewed as showing that a non-homogeneous system of linear equations, where the number of equations equals the number of variables, and this system always has a solution, as long as $\alpha_1, \ldots, \alpha_n$ are distinct. In this question, we will see that this property does not extend to polynomials in a larger number of variables.

   Show that for every $n \geq 2$, there exists a set of points $\{(\alpha_i, \beta_i, \gamma_i) : i \in \{1, 2, \ldots, t\}\} \subseteq \mathbb{F}^3$ with $(\alpha_i, \beta_i) \neq (\alpha_j, \beta_j)$ for all $i \neq j$ and $t \leq \binom{n+2}{2}$, such that for every bivariate polynomial $P(x, y) \in \mathbb{F}[x, y]$ of total degree at most n,

$$\exists i \in \{1, 2, \ldots, t\}, \quad P(\alpha_i, \beta_i) \neq \gamma_i \,.$$

   Observe that if we view this problem linear algebraically, as we did with the univariate case, we again get a non-homogeneous system of linear equations in the coefficients of P, where the number of variables is at most the number of constraints, but unlike the univariate case, this system does not always have a solution.