CS-E4500 Problem Set 6

Jaan Tollander de Balsch - 452056

March 8, 2019

Problem 1

Let F be a field with at least q elements.

(a)

Let $f, \tilde{f} \in F[x]$ be polynomials of degree at most d. Show that if $f \neq \tilde{f}$ then a uniform random $\xi \in F$ satisfies $f(\xi) \neq \tilde{f}(\xi)$ with probability at least $1 - d/q$.

By assuming that the polynomials are not equal $f \neq \tilde{f}$ the polynomial $g = f - \tilde{f}$ is nonzero. The roots ξ of the polynomial g satisfy $f(\xi) = \tilde{f}(\xi)$. Since g is nonzero it has at most d distinct roots $\xi \in F$. Therefore there is at most a probability d/q of picking a uniform random value ξ in F such that it satisfies $f(\xi) = \tilde{f}(\xi)$. Equivalently, there is at least $1 - d/q$ probability that ξ satisfies $f(\xi) \neq \tilde{f}(\xi)$.

(b)

Let $a, b, c \in F[x]$ be three polynomials, each of degree at most d and each given as a sequence of coefficients. Present a randomized test that verified $c = ab$ and uses $O(d)$ operations in F. If $c = ab$ the test must accept with probability 1; if $c = ab$ the test must reject with probability at least $1 - d/q$.

The probability is guaranteed by results from (a).

Evaluate the polynomials a, b, c at uniform random $\xi \in F$ using the Horner's rule. Horner's rule uses $O(d)$ operations in F and assumes constant $O(1)$ complexity for addition and multiplication in F. Then compute $b(\xi)c(\xi)$ and do the comparison $a(\xi) = b(\xi)c(\xi)$ which reduces to addition $a(\xi) - b(\xi)c(\xi) = 0$. Therefore the total number of operations required in for the test is $O(d) + O(1) + (1) = O(d)$.

Problem 2

Let A, B, C be three $n \times n$ matrices with entries in a field F. Present a randomized algorithm that tests whether $C = AB$ using $O(n^2)$ operations in F. When $C = AB$, your algorithm must always assert that $C = AB$. When $C \neq AB$, your algorithm must assert that $C \neq AB$ with probability at least $1/2$.

We have the following equality

$$\begin{aligned} C &= AB \\ Cx &= (AB)x \\ Cx &= A(Bx) \end{aligned}$$

where $x \in F^n$ is a vector. Using this form of the equality, the equality testing will only require $O(n^2)$ operations since the product of $F^{n \times n}$ matric and F^n vector has dimension of F^n and uses $O(n^2)$ operations in F (unlinke naive matrix multiplication, which uses $O(n^3)$).

By assuming $C \neq AB$ we have nonzero matrix $D = (C - AB) \neq \mathbf{0}$. Let $x \in \Omega = \{0, 1\}^n \subseteq F^n$ be a binary vector. Then the probability that we choose a uniform random x such that $Dx \neq 0$ is the same as the probability that $Dx = 0$ due to symmetry.

Proof: Let the star $\star$ represent an element that is either 0 or 1. Let $x \in \Omega$ be a vector such that $Dx \neq 0$. Then x consists of elements 1 and $\star$. Then we can construct a vector $y \in \Omega$ such that $Dy = 0$ by substituting all 1 with 0, but keeping stars $\star$ as stars. As an example:

$$Dx = \begin{bmatrix} \varepsilon & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ \star \end{bmatrix} \neq \mathbf{0} \text{ then } Dy = \begin{bmatrix} \varepsilon & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ \star \end{bmatrix} = \mathbf{0}$$

where $\varepsilon \neq 0$.

Because there is a vector y for every vector x this means that the probabilities are equal

$$P(Dx = 0) = P(Dx \neq 0).$$

Also, probability theory gives us

$$P(Dx = 0) + P(Dx \neq 0) = 1.$$

Therefore the probability

$$P(Dx \neq 0) = \frac{1}{2}.$$

Problem 3

Problem 4

Suppose you have two $x \times x$ matrices, X and Y, with entries in a finite field F with at least four elements. You want to delegate the task of computing the product matric XY to your three friends Alice, Bob and Charlie so that none of your three friends individually gains any information about the matrices X and Y other than the size parameter n. Describe a protocol that employs Alice, Bob, and Charlie to help you so that you obtain the product matrix XY without you yourself putting in more work than $O(n^2)$ operations in F. You can assume you have a subroutine that returns independent uniform random elements of F.

Delegated matrix multiplication can be constructed using evaluation interpolation duality and secret sharing (Gathen and Gerhard 2013, chap. 5.1 - 5.3).

Lets denote the entries of matrix $X = [x_{i,j}]$ and $Y = [y_{i,j}]$ and their product $XY = Z = [z_{i,j}]$ where $i = 1, 2, ..., n$ and $j = 1, 2, ..., n$ denote the indices of the $n \times n$ matrices.

Let the matrices X and Y be the secrets. The number of shares needed to recover the secret is set to be $k = 2$ such that no one can individually gain any information about the secrets. A consequence of this is that the polynomials f and g used for constructing the secrets will have a degree of $\deg f = \deg g = k - 1 = 1$.

Each secret is split into $s = 3$ shares. This is because the matrix product does multiplication on the polynomials used for constructing the secrets and therefore the resulting polynomial is of degree $\deg fg = \deg f + \deg g = 1 + 1 = 2$ which requires $\deg fg + 1 = 3$ points in order to recover it using interpolation.

The shares can be created the following way:

1) Let $\xi_1, \xi_2, \xi_3 \in F$ be distinct and nonzero.
2) Select elements $\varphi_{i,j}, \rho_{i,j} \in F$ independently and uniformly at random.
3) Let $f_{i,j}(x) = x_{i,j} + \varphi_{i,j}x \in F[x]$ and $g_{i,j}(x) = y_{i,j} + \rho_{i,j}x \in F[x]$.
4) Let $X_k = [f_{i,j}(\xi_k))]$ and $Y_k = [g_{i,j}(\xi_k))]$ for $k = 1, 2, 3$.
5) The shares are (ξ_1, X_1, Y_1), (ξ_2, X_2, Y_2), and (ξ_3, X_3, Y_3).

Each polynomial evaluation has two operations in F and they are evaluated in total $2n^2$ times, therefore, there are $2 \cdot 2n^2 = O(n^2)$ operations in F.

These shares are now handed to the friends who perform the matrix multiplications $Z_1 = [z_{i,j,1}] = X_1Y_1$, $Z_2 = [z_{i,j,2}] = X_2Y_2$, and $Z_3 = [z_{i,j,3}] = X_3Y_3$, and then send the results $(\xi_1, Z_1,)$ (ξ_2, Z_2) and (ξ_3, Z_3) back. This part counts as zero operations because the computation is done by the friends.

The polynomial $h_{i,j}$ is obtained by using interpolation on the points $(\xi_1, z_{i,j,1}), (\xi_2, z_{i,j,2}), (\xi_3, z_{i,j,3})$ and then evaluating the polynomial at zero to recover the entries $h_{i,j}(0) = z_{i,j}$ of the product $Z = [z_{i,j}]$.

Proof: $h_{i,j}(0)$ recovers the matrix multiplication

$$\begin{aligned} h_{i,j}(0) &= \sum_{r=1}^{n} f_{i,r}(0) g_{r,j}(0) \\ &= \sum_{r=1}^{n} x_{i,r} y_{r,j} \\ &= z_{i,j} \end{aligned}$$

The interpolation takes a constant amount of operations in F and its done n^2 times, therefore, the total amount of operations is $O(n^2)$. Therefore the total amount of operations for the whole algorithm is $O(n^2)$.

References

Gathen, Joachim von zur, and Jurgen Gerhard. 2013. *Modern Computer Algebra*. 3rd ed. New York, NY, USA: Cambridge University Press.