CS-E4500 Advanced Course in Algorithms (5 cr)

Problem Set 6 *Spring 2019*

1. Randomized polynomial identity testing. 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 $1 - d/q$.

   (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 verifies $c = ab$ and uses $O(d)$ operations in F. If $c = ab$ the test must accept with probability 1; if $c \neq ab$ the test must reject with probability at least $1 - d/q$.

   Hints: For part (a), recall what we know about low-degree polynomials. For part (b), reduce to part (a) and carefully justify that your algorithm uses $O(d)$ operations in F.

2. Testing a matrix product. 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 $1/2$.

   Hints: Select a uniform random $x \in \{0,1\}^n \subseteq F^n$. Study the probability that $Cx \neq A(Bx)$ when $C \neq AB$.

3. Evaluation algorithm for the #CNFSAT proof polynomial $P_{\mathscr{C}}$. Let $\mathscr{C}$ be a collection of clauses $C_1, C_2, \ldots, C_m$ over n variables $x_1, x_2, \ldots, x_n$ taking values in $\{0,1\}$. Present detailed pseudocode for an algorithm that, given as input $\mathscr{C}$, a prime q with $2^{n/2+2}mn \leq q \leq 2^{n/2+3}mn$, and $\xi \in \mathbb{F}_q$, computes the value $P_{\mathscr{C}}(\xi) \in \mathbb{F}_q$ in time $O(2^{n/2}(mn)^c)$ for some constant $c > 0$. Carefully justify the running time of your algorithm. You may use the near-linear-time toolbox for univariate polynomials and algorithms for modular arithmetic in $\mathbb{F}_q$ as subroutines without detailed pseudocode, but make sure that you specify with care the input to each subroutine.

   Hints: The polynomial $P_{\mathscr{C}} \in \mathbb{F}_q[x]$ is defined in the lecture slides. Observe that your algorithm needs to work for an arbitrary $\xi \in \mathbb{F}_q$, not only for $\xi \in \{0,1\}$. Also observe that the given input is $\mathscr{C}$, q, and ξ. In particular, the polynomials $a_1, a_2, \ldots, a_{n/2}$ need to be constructed inside your algorithm.

4. Delegating matrix multiplication. Suppose you have two $n \times n$ 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 matrix 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.

   Hints: Recall Shamir's secret sharing. Extend each matrix X, Y to a matrix whose entries are polynomials of degree at most one with coefficients in F, where the constant of each polynomial is the original matrix entry. Have Alice, Bob, and Charlie each multiply a pair of $n \times n$ matrices $X^{(\mathrm{A})}, Y^{(\mathrm{A})}$, $X^{(\mathrm{B})}, Y^{(\mathrm{B})}$, and $X^{(\mathrm{C})}, Y^{(\mathrm{C})}$ with entries in F. Recover the product matrix XY by interpolation from the products $X^{(\mathrm{A})}Y^{(\mathrm{A})}$, $X^{(\mathrm{B})}Y^{(\mathrm{B})}$, and $X^{(\mathrm{C})}Y^{(\mathrm{C})}$ that Alice, Bob, and Charlie supply to you. Carefully justify that each of your friends on his or her own does not gain any information about X and Y other than the size parameter n.

Deadline and submission instructions. This problem set is due no later than Sunday 10 March 2019, 20:00 (8pm), Finnish time. Please submit your solutions as a single PDF file via e-mail to the lecturer (`petteri.kaski(atsymbol)aalto.fi`). Please use the precise title

```
CS-E4500 Problem Set 6: [your-student-number]
```

with "`[your-student-number]`" replaced by your student number. For example, assuming that my student number is 123456, I would carefully title my e-mail

```
CS-E4500 Problem Set 6: 123456
```

and attach to the e-mail a single PDF file containing my solutions. Please note that the submissions are automatically processed and archived, implying that failure to follow these precise instructions may result in your submission not being graded.