Prof. Friedrich Eisenbrand
Alfonso Cevallos

Discussions from: April 21, 2015

Computer Algebra

Spring 2015

Assignment Sheet 5

Exercises marked with a $\star$ can be handed in for bonus points. Due date is May 5th.
Remarks: If you hand in the homework by e-mail, make sure to type as subject "Computer Algebra HW4". The problems are interconnected, but for each problem you can assume that the results from the previous exercises are known to be true. Finally, remember to consider bit-complexities (so the basic operations are not performed in constant time).

This week we will explore in depth the modular DFT (discrete Fourier transform), where we work over a ring $\mathbb{Z}_M$. From the previous weeks we have gathered the tools necessary to understand the correctness, and analyse the complexity of this algorithm. Theoretically, it is more involved than working over the complex field, but it is simpler (and more elegant) from a computational point of view, as we don't have to deal with the problem of rounding irrational values.

Exercise 1
Recall the definition of a primitive n-th root of unity.

1. Prove that $e^{\frac{2\pi i}{8}}$ is a primitive 8-th root of unity in $\mathbb{C}$, and 3 is a primitive 6-th root of unity in $\mathbb{Z}_7$.
2. Does $\mathbb{Z}_8$ have a primitive square root of unity?
3. Prove that if ω is an n-root of unity, for n even, then ω^2 is an $\frac{n}{2}$-th root of unity.

Exercise 2 ($\star$)
Consider the ring $\mathbb{Z}_M$, where $M = 2^L + 1$, for some natural number L.

1. Let $K = 2^k$ divide L. Prove that $\omega = 2^{L/K}$ is a primitive $2K$-th root of unity.
2. Notice that addition and subtraction in $\mathbb{Z}_M$ can be done with $O(L)$ bit operations. Prove that we can also compute the product $a \cdot 2^j$ in $O(L)$ bit operations, where $a \in Z_M$ and $1 \leq j \leq L$ (this is not just shifting the bit-list to the left, but a little bit more).
3. Let $f(x)$ and $g(x)$ be two polynomials in $\mathbb{Z}_M[x]$, of degree at most $2K-1$ (for K defined as above). Prove that their product $f \cdot g \in \mathbb{Z}_M[x]$ can be computed with $O(KL\log K + KM(L))$ bit operations, where $M(s)$ is the bit complexity of s-bit integer multiplication.

Exercise 3
You want to multiply two n-degree polynomials f and g in $\mathbb{Z}[x]$, using the modular DFT approach. So you map the polynomials into $\mathbb{Z}_M[x]$ via the canonical homomorphism, for a conveniently chosen and large enough M. Once in $\mathbb{Z}_M[x]$, they are multiplied using the modular FFT algorithm, and from this product, the original product in $\mathbb{Z}[x]$ needs to be reconstructed.

1. Let B be an upper bound on the absolute values of the coefficients of f and g. Determine a positive integer M such that the reconstruction of the product $f \cdot g \in \mathbb{Z}[x]$, from $f \cdot g \in \mathbb{Z}_M[x]$, is guaranteed to be unique. Try to choose an M that is as small as possible. (Can you prove that your choice of M is tight?)

2. Derive an upper bound on the bit complexity of the whole process, in terms of n and $size(B)$ (try to make your analysis as tight as possible).

Exercise 4 ($\star$)

Consider the polynomials $f(x) = 5x^3 + 3x^2 - 4x + 3$ and $g(x) = 2x^3 - 5x^2 + 7x - 2$ in $\mathbb{Z}_{17}[x]$.

1. Compute $f \cdot g$ in the standard way.

2. Show that $\omega = 2$ is a primitive 8th root of unity, and compute the inverse $\omega^{-1} = 2^{-1}$.

3. Compute the Vandermonde matrices V_ω and $V_{\omega^{-1}}$, and their product.

4. Perform the modular FFT algorithm on input ω, f and ω, g (show your steps), and compute the product $y = (y_0, \cdots, y_7) = DFT_\omega(f) \cdot DFT_\omega(g)$.

5. Perform the same algorithm on input ω^{-1}, y, and compare your result with part 1.

Exercise 5 ($\star$)

1. Implement the modular FFT algorithm that, given a primitive n-th root of unity ω (where n is a power of 2) and the coefficients $a = (a_0, \cdots, a_{n-1})$, computes $DFT_\omega(a)$.

2. Implement an algorithm that computes the product of two polynomials in $\mathbb{Z}[x]$, using modular DFT. Test it on several polynomials.