Michael Sagraloff **Winter term 2017/18**

Computer Algebra

https://www.mpi-inf.mpg.de/departments/algorithms-complexity/teaching/winter17/comp-alg/

Assignment sheet 4 due: Monday, November 20

Exercise 1: Polynomial Evaluation (4 points)

Let x_0 be an integer of length less than ℓ and $f(x) = \sum_{j=1}^{k} a_{i_j} \cdot x^{i_j} \in \mathbb{Z}[x]$ a (so called *sparse*) polynomial with $0 \leq i_j \leq n$ for all j and $|a_{i_j}| < 2^L$ for all j. Show that one can compute $f(x_0)$ using $\tilde{O}(k \cdot (n\ell + L))$ primitive operations!

Hint: Show first that one can compute x_0^n using $\tilde{O}(n\ell)$ primitive operations.

Exercise 2: Estrin's scheme vs. Horner's scheme (4 points)

You have already seen Horner's scheme for polynomial evaluation. An alternative method is *Estrin's scheme:* To evaluate a polynomial $f(x) = a_n x^n + \cdots + a_0$, let $m := 2^{\lceil \log n \rceil - 1}$ and write f as

$$f(x) = \underbrace{(a_n x^m + a_{n-1} x^{m-1} + \cdots a_m)}_{=:f_H(x)} \cdot x^m + \underbrace{a_{m-1} x^{m-1} + a_{m-2} x^{m-2} + \cdots + a_0}_{=:f_L(x)},$$

where f_H and f_L are polynomials of degree at most m. Recursively evaluate f_H and f_L and reconstruct $f(x) = f_H(x) \cdot x^m + f_L(x)$. (Notice that it suffices to compute the powers $x, x^2, x^4, \ldots$ of x in a preprocessing step.)

Provide and compare complexity bounds for the computation of $f(x_0)$ with Horner's and Estrin's methods, where f is an integer polynomial of degree n with coefficients of length less than L and $x_0 \in \mathbb{Z}$ is an integer of length ℓ.

Exercise 3: Fast bivariate polynomial multiplication (4 points)

Show that two polynomials f and $g \in \mathbb{Z}[x, y]$ of total degree at most n with coefficients of length less than L can be multiplied using $\tilde{O}(n^2 L)$ primitive operations.

Hint: Use Kronecker substitution!

Exercise 4: Fast Integer Multiplication (4 point + 4 bonus points for *)

Let $n = 2^{2^k}$ with $k \in \mathbb{N}$.

(a) Show that $\omega := 8$ is a primitive $2n$-th root of unity in $R := \mathbb{Z}/(2^{3\sqrt{n}}+1)\mathbb{Z}$.

(b) Let $a = a_{n-1}a_{n-2}\ldots a_0$ and $b = b_{n-1}b_{n-2}\ldots b_0$ be two integers of length n. Consider the integer polynomials

$$f(x) := \sum\nolimits_{i=0}^{\sqrt{n}-1} (a_{(i+1)\sqrt{n}-1} \ldots a_{i\sqrt{n}+1}a_{i\sqrt{n}}) \cdot x^i$$
$$g(x) := \sum\nolimits_{i=0}^{\sqrt{n}-1} (b_{(i+1)\sqrt{n}-1} \ldots b_{i\sqrt{n}+1}b_{i\sqrt{n}}) \cdot x^i,$$

and their images $f^* := f \mod (2^{3\sqrt{n}}+1)$ and $g^* := f \mod (2^{3\sqrt{n}}+1)$ in $R[x]$. Show that the coefficients of $h^* = f^* \star_{2\sqrt{n}} g^* \in R[x]$ equal the coefficients of $f \cdot g \in \mathbb{Z}[x]$, and conclude that h can be computed with $O(n \log n)$ arithmetic operations in R.

(c)* Notice that, for computing h^*, we need only $2\sqrt{n}$ *essential* multiplications in R, whereas the remaining multiplications are multiplications by powers of ω. Which complexity bound can you derive for the computation of $a \cdot b$ when using the approach recursively for the essential multiplications?

Hint: You should first prove that each of these essential multiplications can be reduced to a constant number of additions and multiplications of integers of length $\sqrt{n}$.