Albert-Ludwigs-Universität, Inst. für Informatik
Prof. Dr. Fabian Kuhn
H. Ghodselahi, O. Saukh, M. Ahmadi October 21, 2015

Algorithm Theory, Winter Term 2015/16
Problem Set 2

hand in (hard copy or electronically) by 10:15, Thursday November 5, 2015,
tutorial session will be on November 9, 2015

Exercise 1: Multiplication of Polynomials ($2 + 7$ points)

Given a polynomial $p(x)$:

$$p(x) = x^3 - x^2 + 2x + 1$$

- Compute $DFT_4(a)$, where a is the coefficient vector of polynomial p.
- Compute the coefficient representation of $p(x)^2$ by using the FFT algorithm from the lecture.

Remark: Instead of using exact numbers for the point-wise evaluations (which involves irrational numbers) you can also round numbers to, say, 3+ digits after the decimal point. This also reflects what happens in an implemented version of FFT, as exact algebraic evaluations would not lead to an $O(n \log n)$ running time. Those unfamiliar with complex numbers should ask fellow students for some help - calculating roots of unity and multiplying two complex numbers is all you need for this exercise.

Exercise 2: Polynomial to the power of k (3 points)

Given a polynomial $p(x)$ of degree n and an integer $k \geq 2$, the goal of this problem is to compute the k^{th} power $p^k(x)$ of $p(x)$ in an efficient way. For simplicity, we assume that k is a power of 2, that is, $k = 2^\ell$ for some integer $\ell \geq 1$.

- Describe an efficient algorithm to compute $p^k(x)$ polynomial using the *Fast Polynomial Multiplication* algorithm from the lecture.
- What is the asymptotic runtime of your algorithm in terms of k and n? Explain your answer.