University of Freiburg
Dept. of Computer Science
Prof. Dr. F. Kuhn
S. Faour, M. Fuchs, A. Malyusz

Algorithm Theory
Exercise Sheet 2

Due: Friday, 3rd of November 2023, 10:00 am

Exercise 1: Faster Polynomial Multiplication *(14 Points)*

Let $p(x) := -3x^2 + x + 6$ and $q(x) := 2x^2 + 4$. The goal is to compute $p(x) \cdot q(x)$ with the help of the FFT algorithm. Please, make use of the following sketch:

1. Illustrate the **divide** procedure of the algorithm (for both functions p and q). More precisely, for the i-th divide step (with focus on $p(x)$), write down all the polynomials p_{ij} for $j \in \{0, ..., 2^i - 1\}$ that you obtain from further dividing the polynomials from the previous divide step $i - 1$ (we define $p_{00} := p$, and the first split is into p_{10} and p_{11} and so on...).
2. Illustrate the **combine** procedure of the algorithm (for both functions p and q). That is, starting with the polynomials of the smallest degree as base cases, compute the DFT of p_{ij} (respectively q_{ij}) bottom up with the recursive formula given in the lecture. The recursion stops when $DFT_8(p_{00})$ (respectively $DFT_8(q_{00})$) is computed i.e., we know the function's values at the (8-th) roots of unity.
3. **Multiply** the polynomials. More specific, give the point value representation of $p(x) \cdot q(x)$, i.e., $(w_8^0, y_0), (w_8^1, y_1), \ldots, (w_8^7, y_7)$.
4. Use the **inverse** DFT procedure from the lecture to get the final coefficients for $p(x) \cdot q(x)$. To do that efficiently, first compute the $DFT_8(f)$ where $f(x) := y_0 + y_1 \cdot x + ... + y_7 \cdot x^7$ and then compute the coefficients a_k for $k \in \{0, 1, ..., 7\}$ of $p(x) \cdot q(x)$ (using that $a_k = 1/8 \cdot f(w_8^{-k})$).

Write down all intermediate results to get partial points in the case of a typo.

Exercise 2: FFT Application *(6 Points)*

Let A, B be two sets of integers between 0 and n i.e., $A, B \subseteq \{0, 1, 2, \ldots, n\}$. We define two random variables X_A and X_B, where X_A is obtained by choosing a number uniformly at random from A and X_B is obtained by choosing a number uniformly at random from B. We further define the random variable $Z := X_A + X_B$. Note that Z can take values in the range $0, \ldots, 2n$.
Give an $O(n \log n)$ algorithm to compute the distribution of Z. Hence, the algorithm should compute the probability $P(Z = z)$ for all $z \in \{0, \ldots, 2n\}$. Note that $\sum_{z=0}^{2n} P(Z = z) = 1$. You can use the algorithms of the lecture as a black box. State the correctness of your algorithm and also explain the runtime!