University of Freiburg
Dept. of Computer Science
Prof. Dr. F. Kuhn
P. Bamberger

Algorithms Theory
Sample Solution Exercise Sheet 2

Exercise 1: Convolution *(11 Points)*

Compute the convolution of the vectors $a = (5, 8, -2, 3)$ and $b = (-9, 4, -1)$ using the algorithm for polynomial multiplication from the lecture. Document all computation steps for evaluation, point-wise multiplication and interpolation.

Sample Solution

Let $p^a(x) = 3x^3 - 2x^2 + 8x + 5$ and $p^b(x) = 0x^3 - x^2 + 4x - 9$. We want to compute the coefficients $c_0, \dots, c_6$ of the polynomial $p^a(x) \cdot p^b(x)$.

Evaluation: $p^a(x) \cdot p^b(x)$ has maximum degree 5, i.e., is defined by 6 point-value pairs. We evaluate the polynomials at 8 points (the next power of 2), namely at the 8th roots of unity

$$X = \{\omega_8^0, \omega_8^1, \omega_8^2, \omega_8^3, \omega_8^4, \omega_8^5, \omega_8^6, \omega_8^7\} = \{1, \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}}i, i, -\frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}}i, -1, -\frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}}i, -i, \frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}}i\}$$

using the FFT algorithm. For $p^a(x)$ we obtain the following recursion tree

$$p^a(x) = 3x^3 - 2x^2 + 8x + 5$$

$$p_0^a(x) = -2x + 5 \qquad p_1^a(x) = 3x + 8$$

$$p_{00}^a(x) = 5 \qquad p_{01}^a(x) = -2 \qquad p_{10}^a(x) = 8 \qquad p_{11}^a(x) = 3$$

We proceed as follows: First we evaluate the polynomials on the lowest level for all $x \in X^4 = \{1, -1\}$ (which is already done as these polynomials are constant), then we evaluate $p_0^a(x)$ and $p_1^a(x)$ for all $x \in X^2 = \{1, i, -1, -i\}$ using the combine rules $p_0^a(x) = p_{00}^a(x^2) + x \cdot p_{01}^a(x^2)$ and $p_1^a(x) = p_{10}^a(x^2) + x \cdot p_{11}^a(x^2)$ which yields

- $p_0^a(1) = 3$
- $p_0^a(i) = 5 - 2i$
- $p_0^a(-1) = 7$
- $p_0^a(-i) = 5 + 2i$
- $p_1^a(1) = 11$

- $p_1^a(i) = 8 + 3i$
- $p_1^a(-1) = -5$
- $p_1^a(-i) = 8 - 3i$

and finally we compute $p^a(x)$ for all $x \in X$ using $p^a(x) = p_0^a(x^2) + x \cdot p_1^a(x^2)$

- $p^a(\omega_8^0) = 14$
- $p^a(\omega_8^1) = 5 + \frac{5}{\sqrt{2}} + \left(\frac{11}{\sqrt{2}} - 2\right) i$
- $p^a(\omega_8^2) = 7 + 5i$
- $p^a(\omega_8^3) = 5 - \frac{5}{\sqrt{2}} + \left(\frac{11}{\sqrt{2}} + 2\right) i$
- $p^a(\omega_8^4) = -8$
- $p^a(\omega_8^5) = 5 - \frac{5}{\sqrt{2}} - \left(\frac{11}{\sqrt{2}} + 2\right) i$
- $p^a(\omega_8^6) = 7 - 5i$
- $p^a(\omega_8^7) = 5 + \frac{5}{\sqrt{2}} + \left(\frac{11}{\sqrt{2}} - 2\right) i$

The same procedure done for $p^b(x)$ yields

- $p^b(\omega_8^0) = -6$
- $p^b(\omega_8^1) = -9 + \sqrt{8} + (\sqrt{8} - 1)i$
- $p^b(\omega_8^2) = -8 + 4i$
- $p^b(\omega_8^3) = -9 - \sqrt{8} + (\sqrt{8} + 1)i$
- $p^b(\omega_8^4) = -14$
- $p^b(\omega_8^5) = -9 - \sqrt{8} - (\sqrt{8} + 1)i$
- $p^b(\omega_8^6) = -8 - 4i$
- $p^b(\omega_8^7) = -9 + \sqrt{8} + (-\sqrt{8} - 1)i$

Point-wise multiplication: We obtain the pairs $(\omega_8^0, y_0), \ldots, (\omega_8^7, y_7)$ with

- $y_0 = -84$
- $y_1 = -59 - 3\sqrt{2} + (45 - 46\sqrt{2})i$
- $y_2 = -76 - 12i$
- $y_3 = -59 + 3\sqrt{2} + (-45 - 46\sqrt{2})i$
- $y_4 = 112$
- $y_5 = -59 + 3\sqrt{2} + (45 + 46\sqrt{2})i$
- $y_6 = -76 + 12i$
- $y_7 = -59 - 3\sqrt{2} + (-45 + 46\sqrt{2})i$

Interpolation: Let $q(x) = y_0 + y_1 x + \cdots + y_7 x^7$ with y_i as above. We obtain

$$(c_0, \ldots, c_7) = \frac{1}{n} \left(q(\omega_8^0), q(\omega_8^7), q(\omega_8^6) \ldots, q(\omega_8^1)\right)$$

The computation of $q(\omega_8^i)$ can be done again with the FFT algorithm. We obtain

$$(c_0, \ldots, c_7) = (-45, -52, 45, -43, 14, -3)$$

Exercise 2: Scheduling *(11 Points)*

Given n jobs of lengths $t_1 \dots, t_n$ with **one deadline** $d \geq 0$, we want to schedule these jobs such that the **average lateness** is minimized. That is, for each job i we want to find a start and finishing time $0 \leq s(i) \leq f(i)$ with $f(i) - s(i) = t_i$ such that the intervals $[s(i), f(i)]$ are pairwise non-overlapping and the average over all $L(i) = \max\{0, f(i) - d\}$ is minimal (overlapping of start- and endpoints is allowed).

Describe a greedy algorithm for this problem and prove that it computes an optimal solution.

Sample Solution

We schedule the jobs by length, starting with the shortest and ending with the longest (and of course do not leave any space between two jobs). This minimizes the sum of all latenesses (and hence the average lateness). We proof it with an exchange argument. Let O be an optimal solution. We transfer O to a greedy solution without increasing the total lateness (if the job lengths are not pairwise distinct, there are different greedy solutions). To ease presentation, assume that each job is represented by an integer such that $O = (1, \dots, n)$. If O is not a greedy solution, there must be jobs i and $i+1$ with $t_i > t_{i+1}$. We exchange jobs i and $i+1$ and compare the old and new finishing times of all jobs:

$$f_{new}(i) = \sum_{j<i} t_j + t_{i+1} + t_i = \sum_{j \leq i+1} t_j = f_{old}(i+1)$$

and

$$f_{new}(i+1) = \sum_{j<i} t_j + t_{i+1} < \sum_{j<i} t_j + t_i = f_{old}(i)$$

For the latenesses it follows $L_{new}(i) = L_{old}(i+1)$ and $L_{new}(i+1) \leq L_{old}(i)$. The finishing time and thus the lateness of all other jobs does not change. We obtain

$$\begin{aligned}
\sum_{j=1}^{n} L_{new}(j) &= \sum_{j=1}^{i-1} L_{new}(j) + L_{new}(i) + L_{new}(i+1) + \sum_{j=i+2}^{n} L_{new}(j) \\
&\leq \sum_{j=1}^{i-1} L_{old}(j) + L_{old}(i+1) + L_{old}(i) + \sum_{j=i+2}^{n} L_{old}(j) \\
&= \sum_{j=1}^{n} L_{old}(j)
\end{aligned}$$

So we have seen that exchanging jobs i and $i+1$ did not increase the sum of all latenesses and thus the average lateness did not increase. We proceed this way until the jobs are sorted by length, i.e., we obtain a greedy solution. It follows inductively that the average lateness of this solution is not larger than the one of O and therefore the greedy solution is optimal.

Exercise 3: Matroids *(8 Points)*

We are given a directed weighted graph $G = (V, E)$, where $w : E \to \mathbb{R}^+$ defines weights of the edges. Consider also a function $b : V \to N$ that defines some indegree bound for each node. We would like to find a subset $E' \subseteq E$ of maximum total weight such that every node $u \in V$ has indegree at most $b(u)$ in the graph $G' = (V, E')$. Show that the set of feasible solutions form a matroid and thus, this problem can be solved by using the greedy algorithm for matroids.

Sample Solution

Let $\mathcal{I} := \{E' \subseteq E \mid \deg^{in}_{E'}(v) \leq b(v)\}$ where $\deg^{in}_{E'}(v)$ denotes the indegree of v in (V, E'). We show that $(E, \mathcal{I})$ is a matroid.

- Empty set is independent: $\emptyset \in \mathcal{I}$, because $\deg^{in}_{\emptyset}(v) = 0 \leq b(v)$ for all $v \in V$.
- Hereditary property: Let $A' \subseteq A \in \mathcal{I}$. It follows $\deg^{in}_{A'}(v) \leq \deg^{in}_{A}(v) \leq b(v)$ for all $v \in V$ and hence $A' \in \mathcal{I}$.
- Augmentation/Independent set exchange property: Let $A, B \in \mathcal{I}$ with $|A| > |B|$. Then there must be a node $v \in V$ with $\deg^{in}_{B}(v) < \deg^{in}_{A}(v) \leq b(v)$. Let e be an incoming edge of v with $e \in A \setminus B$. Let $B' := B \cup \{e\}$. It follows $\deg^{in}_{B'}(v) = \deg^{in}_{B}(v) + 1 \leq b(v)$ and hence $B' \in \mathcal{I}$.