Albert-Ludwigs-Universität, Inst. für Informatik
Prof. Dr. Fabian Kuhn
H. Ghodselahi, Y. Maus October 27, 2016

Algorithm Theory, Winter Term 2016/17
Problem Set 2

hand in (hard copy or electronically) by 09:55, Thursday November 10, 2016, tutorial session will be on November 14, 2016

Exercise 1: Almost Closest Pairs of Points (16 points)

In the lecture, we discussed an $\mathcal{O}(n \log n)$-time divide-and-conquer algorithm to determine the closest pair of points. Assume that we are not only interested in the closest pair of points, but in all pairs of points that are at distance at most twice the distance between the closest two points.

a) **(4 points)** How many such pairs of points can there be? It is sufficient to give your answer using big-$\mathcal{O}$ notation.

b) **(12 points)** Devise an algorithm that outputs a list with all pairs of points at distance at most twice the distance between the closest two points. Describe what you have to change compared to the closest pair algorithm of the lecture and analyze the running time of your algorithm.

Exercise 2: Polynomial Multiplication using FFT (10 points)

Let $p(x)$ be a polynomial of degree n and $q(x)$ a polynomial of degree m. If $m = n$ the multiplication algorithm in the lecture (using FFT) yields to $\mathcal{O}(n \log n)$ runtime. Now suppose $n > m$. How can one do the multiplication in $\mathcal{O}(n \log m)$ time?

Exercise 3: Interval Scheduling (14 points)

In the *interval scheduling* problem, we are given a set of intervals each with starting and ending times. The goal is to select a largest possible non-overlapping set of intervals. Let us assume overlaps at the boundaries are fine. In the lecture, we have studied a greedy algorithm called *shortest available interval* for solving the problem. We have seen that the algorithm fails to optimally solve the problem.
Show that the above greedy algorithm returns a set of intervals in which the size of the set is **at least half** of the number of intervals provided by the optimal algorithm.