Karl Bringmann, Erik Jan van Leeuwen **Winter 2016/2017**

Exercises for Algorithms and Data Structures

http://www.mpi-inf.mpg.de/departments/algorithms-complexity/teaching/winter16/algorithms-and-data-structures/

Exercise Sheet 13 Due: **13.2.2017**

The homework must be handed in on Monday before the lecture. You may collaborate with other students on finding the solutions for this problem set, but every student must hand in a writeup in their own words. We also expect you to state your collaborators and sources (books, papers, course notes, web pages, etc.) that you used to arrive at your solutions.

You need to collect at least 50% of all points on all exercise sheets to be admitted to the final exam.

Whenever you are asked to design an algorithm in this exercise sheet you have to give a proof of its correctness as well as an asymptotic upper bound on its worst case running time.

Exercise 1 (*10 points*)

Implement the following primitives in constant time. You can choose any input representation.

a) (5 points) Design an algorithm that, given two lines in the plane, either outputs a point in which the two lines intersect or reports correctly that no such point exists.

b) (5 points) Do the same with "lines" replaced by "line segments".

Exercise 2 (*10 points*)

You are given a set P of n points placed on the real line. Every point $p \in P$ has an associated real weight w_p. Build a data structure that allows to answer queries of the following type: Given an interval I on the real line, output the point in $P \cap I$ with the highest weight. Your data structure should use preprocessing time $O(n \log n)$, space $O(n)$, and query time $O(\log n)$.

Exercise 3 (*10 points*)

Let P be a set of n points and R be a set of n (axis-aligned) rectangles in the plane. We want to determine all pairs (p, r) of a point $p \in P$ and a rectangle $r \in R$ such that p is contained in r. Design a sweep line algorithm that runs in time $O(n \log n + k)$, where k is the output size.

Exercise 4 (*10 points*)

For two points p and q in the plane we say that p dominates q if in both coordinates p is at least as large as q, i.e., $p_x \geq q_x$ and $p_y \geq q_y$. You are given a set P of n points in the plane. Design an algorithm that determines for each point $p \in P$ the number of points in $P \setminus \{p\}$ that p dominates. You get 10 points for time $O(n \log n)$, 8 points for time $O(n \log^2 n)$.

Exercise 5 (*10 BONUS points*)

A *slab* in the plane is the subset of the plane delimited by two parallel lines. The width of such a slab is the distance between the two lines that define it. We define the width of a finite set T in the plane by the smallest width of any slab that contains T.

a) (4 points) Let S be a slab that defines the width of a set T. How many points of T must at least lie on the bounding lines of S?

b) (6 points) Show that the width of a set T of n points in the plane can be determined in $O(n \log n)$ time.