Due Thursday December 19, 23:59, by e-mail (pdf)

Requirements: Assignments have to be typeset in English. They have to be handed in by each student separately, except for exercises marked with (G), which should be solved and handed in by each (project) group. Please include your name on top of the first sheet.

How to describe algorithms: Whenever you are asked to describe an algorithm, you should present three things: the algorithm, a proof of its correctness, and a derivation of its running time.

A careful geometric proof can be quite cumbersome at times, so you are highly encouraged to use intuition and give illustrations whenever appropriate. However, beware of “proof by picture”: a poorly drawn figure might make you ignore special cases that are not apparent in your drawing. Try to keep your answers concise and to the point. Never give detailed code. Only give sufficient detail to convince us that your method and your running time analysis are complete and correct.

Unless stated otherwise, you can assume that objects are in general position, that is, no three points are collinear and no four points are cocircular. You may also assume that any geometric primitive that involves only a constant number of objects which are each of constant complexity can be computed in $O(1)$ time.

Exercise 1: [5 points] Let $I$ be a set of intervals on the real line. Hint: In this exercise you should reuse data structures from the lectures.

(a) Describe a data structure using $O(n \log n)$ storage that can efficiently report the intervals that are completely contained in a query interval $[x, x']$. What is the query time?

(b) Describe a data structure using $O(n)$ storage that can efficiently report the intervals that contain a query interval $[x, x']$. What is the query time?

Exercise 2: [5 points] In the lecture we consider the query problem: Given a set of disjoint line segments in the plane, report the segments intersecting a vertical line segment $s$.

(a) Now suppose we instead want to report the segments intersecting a vertical ray running from a point $q$ vertically upwards. Describe a data structure for this problem that uses $O(n \log n)$ storage and that answers queries efficiently (i.e., faster than the data structure from the lecture).

(b) Reduce the storage requirement for the case that we only want to report the first segment hit by the query ray.

Exercise 3(G): [4 points] In this exercise you explore the trade-off between query time and storage for range trees on $n$ points in 2d. For this we could bring down the storage requirement for a range tree by storing associated structures only with a subset of the nodes in the main tree.
(a) Suppose that only the nodes in every $i$th level have an associated structure. Show how the query algorithm can be adapted to answer queries correctly.

(b) Suppose $i = \log \log n$. Analyze the storage requirements and query time for such a data structure.

(c) Suppose we just want to achieve a query time of $O(n^\varepsilon + k)$ for a constant $0 < \varepsilon < 1$, where $k$ is the size of the output. Analyze the storage requirements.