Computational Geometry — Fall 2017/18 — Dan Halperin

Assignment no. 4

due: January 15th, 2018

Exercise 4.1 Give an example of a set of n points in the plane, and a query rectangle for which the number of nodes of the kd-tree visited is $\Omega(\sqrt{n})$.

Exercise 4.2 The algorithm we saw in class for searching in a kd-tree (where the search is guided by comparing the *region* of a node with the query region) can also be used when querying with ranges other than rectangles. For example, a query is answered correctly if the range is a triangle.
(a) Show that the query time for range queries with triangles is linear in the worst case, even if no points are reported at all. Hint: Choose all the input points to lie on the line $y = x$.
(b) Suppose that a data structure is needed that can answer triangular range queries but only for triangles whose edges are horizontal, vertical or have slope $+1$ or -1. Devise a linear-size data structure that answers such queries in $O(n^{3/4} + k)$ time, where k is the number of points to be reported. Hint: Choose 4 coordinate axes in the plane and use a "4-dimensional" kd-tree.
(c) Improve the query time to $O(n^{2/3} + k)$.

Exercise 4.3 Given a star-shaped polygon P with n vertices, show that after expected $O(n)$ preprocessing time, one can determine whether a query point lies in P in worst-case $O(\log n)$ time.

Exercise 4.4 Design an algorithm with running time $O(n \log n)$ for the following problem: Given a set P of n points, determine a value of $\varepsilon > 0$ such that the shear transformation $\Phi : (x, y) \to (x + \varepsilon y, y)$ does not change the order (in x-direction) of points with unequal x-coordinates.

Exercise 4.x, bonus Give a randomized algorithm to compute all pairs of intersecting segments in a set of n line segments in the plane in expected time $O(n \log n + k)$, where k is the total number of intersections among the segments.

Exercise 4.5 Let P be a set of n points in the plane. Give an $O(n \log n)$ time algorithm to find for each point p in P another point in P that is closest to p.

Exercise 4.6 Do the breakpoints of the beach line in Fortune's algorithm always move downwards when the sweep line moves downwards? Prove this or give a counterexample.