Computational Geometry — Spring 2012 — Dan Halperin

Assignment no. 3

due: May 14th, 2012

Exercise 3.1 A simple polygon P is called *star-shaped* if it contains a point q such that for any point p in P the line segment $\overline{pq}$ is contained in P. Give a linear time algorithm to decide whether a simple polygon is star-shaped.

Exercise 3.2 Describe in detail the procedure UnboundedLP3, which accepts n halfspaces in three-dimensional space, and an objective function. The procedure either finds that the induced LP is infeasible, or that the LP is bounded in which case UnboundedLP3 outputs three witnesses to this fact, or outputs a ray (in three-dimensional space) such that as we proceed away from the ray's terminus, the objective function grows. Show that the procedure runs in time $O(n)$.

Exercise 3.3 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 3.4 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 3.5 Given a star-shaped polygon P with n vertices, show that after $O(n)$ preprocessing time, one can determine whether a query point lies in P in $O(\log n)$ time.

Exercise 3.6 (optional) Give a randomized algorithm to compute all pairs of intersecting segments in a set of n line segments in expected time $O(n \log n + A)$, where A is the number of intersecting pairs.