Computational Geometry — Spring 2019 — Dan Halperin

Assignment no. 5

due: June 24th, 2019

Exercise 5.1 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 5.2 Give an efficient algorithm to compute the medial axis of a convex polygon. Analyze the running time of your algorithm.

Exercise 5.3 (revised on 11/6/2019) To complete the sweep-line algorithm for computing the Voronoi diagram of points in the plane, write a procedure to wrap up a *valid* DCEL representation of the diagram within a sufficiently large bounding box. The box should contain all the sites and all the Voronoi vertices. The necessary information is available in the incomplete DCEL computed throughout the sweep as well as the status structure after all the events have been handled.

Exercise 5.4 Let G be the search DAG constructed by the trapezoidal RIC algorithm for point location in the planar subdivision induced by n pairwise interior-disjoint line segments. Let L denote the length of the longest query path in G, and D denote the depth of the DAG. Prove that the $\Omega(n/\log n)$ lower bound on the ratio D/L is tight.

Exercise 5.5 Describe how to use kd-trees for efficiently finding the k-nearest neighbors, where k is a positive integer given as part of the query. Describe the algorithm in detail, as well as auxiliary data structures if needed. How much storage is required by the structure? No need to analyze the complexity of the search.

Exercise 5.6 (optional, bonus) Let T be the search structure constructed by the trapezoidal RIC algorithm for point location in the planar subdivision induced by n pairwise interior-disjoint line segments, but this time **without merges**. Namely, when we insert a new segment we split existing trapezoids as in the standard construction, but we do not merge trapezoids. Show that the expected number of leaves in T is $O(n \log n)$.