Computational Geometry — Spring 2020 — Dan Halperin

Assignment no. 5 (optional)

due:[1] June 24th, 2020

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 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 5.3 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 $\Phi : (x, y) \rightarrow (x + \varepsilon y, y)$ does not change the order (in x-direction) of points with unequal x-coordinates.

Exercise 5.4 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 5.5 A Euclidean minimum spanning tree (EMST) of a set P of points in the plane is a tree of minimum total (Euclidean) edge length connecting all the points.
(a) Prove that the set of edges of a Delaunay triangulation of P contains an EMST for P.
(b) Use this result to give an $O(n \log n)$ time algorithm to compute an EMST for P.

[1] The assignment will be checked but not graded. It will be checked only if you submit it by the due date.