Final – 250 points 600.488 Computational Geometry May 7, 1997

1. Please define each of the following terms using one or two sentences:

   (a) ham-sandwich cut

   (b) trapezoidal decomposition

   (c) the 2-dimensional linear programming problem

2. Delaunay triangulation.

   (a) Draw, as best you can, the Delaunay triangulation of the following set of points:

   $$(2, 2)$$
   $$(2, 3)$$
   $$(1, 4)$$
   $$(1, 0)$$
   $$(5, 1)$$
   $$(3, 4)$$
   $$(5, 5)$$

   (b) Please briefly describe an $O(n \log n)$-time method for constructing the Delaunay triangulation of n points in the plane.

3. Draw, as best you can, the arrangement of the following set of lines:

   $$\begin{aligned} 2x + y &= 1 \\ 2x - y &= 1 \\ x + 3y &= 3 \\ x + 2y &= 2 \\ -x - y &= 1 \end{aligned}$$

   Also, please briefly describe an $O(n^2)$-time method for constructing the arrangement of n lines in the plane.

4. What does it mean for an algorithm to be *output-size sensitive*? Describe an output-size sensitive method for constructing all pairwise intersections between a set of n line segments in the plane. What is the running time of this method?

5. Sketch a fast method for constructing the convex hull of n points in $\mathbf{R}^3$ (i.e., 3-dimensional Euclidean space). How can your method be used to construct the Delaunay triangulation of n points in the plane?

6. Explain how the size of the lower envelope of a set of n line segments in the plane is related to an order-3 Davenport-Schinzel sequence. Describe an efficient algorithm for constructing such a lower envelope. What is the running time of this method?

   NOTE: For the remainder of this exam you may assume that you have a subroutine for any problem we discussed in class, provided you can correctly characterize its performance bounds.

7. Let P and Q be two polygonal chains in the plane. Describe an $O(n \log n)$-time algorithm for determining if P and Q can be separated by a line (with P entirely on one side and Q entirely on the other).

8. A polygon P is said to be *star shaped* if there is a point q inside the polygon that can see the entire polygon. Describe a simple data structure for quickly determining if a query point q is inside a given n-edge simple polygon P. (You should assume that there will be many possible query points, so query time should be minimized.) What is the preprocessing time, space complexity, and query time for your method?

9. Say that a point p in the plane *dominates* a point q in the plane if the x- and y-coordinates of p are both greater than the respective x- and y-coordinates of q. Given a set S of n points in the plane, describe an efficient algorithm for finding all those points in S that are not dominated by any other point in S. What is the running time of this method?

10. Let P be a simple polygon, consisting of n edges. The *kernel* of P is the set of all points inside P that can see all of P. Describe an $O(n \log n)$ algorithm for constructing the kernel of P.