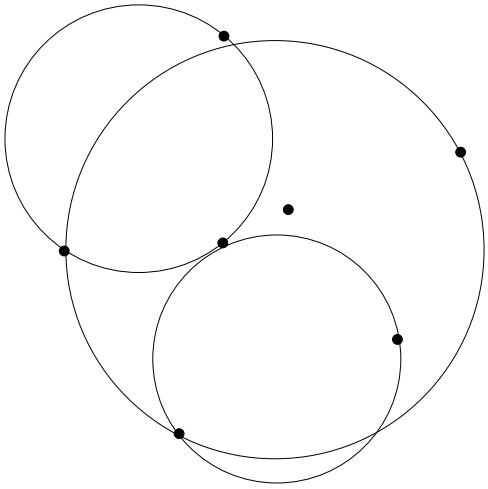
COMPUTATIONAL GEOMETRY
Practice Final

Statistics: $n = 36$, $\mu = 67.6$, median 67, $\sigma = 12.8$; score range: 42–93

(1). [12 points] For the set S of 7 points shown below, do the following:

(a). Draw the (Euclidean) Delaunay diagram. In order to assist you in making some decisions (in case you do not have a compass with you), I have drawn a few circles.



(b). Sketch also the Voronoi diagram.

(c). Draw the furthest-site Delaunay diagram (dual to the furthest-site Voronoi diagram)

(2). [30 points] For each of the computations below indicate how efficiently one can perform the calculation, in terms of $O(...)$ notation (e.g., $O(n)$, $O(\log n)$, $O(n^2)$, $O(n \log n)$, $O(k \log n)$, etc). Try to give the best (lowest) upper bound possible. **For full credit, you must give a brief justification of each answer. (One sentence should suffice.)**

(a). Given an arbitrary set of n triangles in the plane (in general position), report all k points of intersection between pairs of boundary segments among the triangles.

(b). Compute the Euclidean minimum spanning tree of n points in the plane.

(c). Given a set S of n points in the plane, determine if the convex hull, $CH(S)$, is a triangle.

(d). Given a set S of n points in the plane, determine if one of them (say, $p_1 \in S$) is a vertex of the convex hull, $CH(S)$.

(e). Given a set $\mathcal{L}$ of n lines, build a data structure to support efficient queries of the form: How many sides does the cell containing query point q have in the arrangement induced by $\mathcal{L}$? State the preprocessing time, storage space, and query time.

(i). Preprocessing time is $O($ $)$
(ii). Storage space (memory usage) is $O($ $)$
(iii). Query time is $O($ $)$

(f). Given a list of points, $(p_1, p_2, \ldots, p_n)$, in the plane, defining a polygonal chain (with edges $p_i p_{i+1}$), determine if the chain crosses itself at any point.

(3). [10 points] We want to solve the following query problem: Given a set S of n disjoint line segments in the plane, determine the first segment that is stabbed by a vertical ray running from a point $q = (q_x, q_y)$ vertically upwards to infinity. Describe briefly a data structure and a method for this problem. Try to be as efficient as possible in both space and query time.

(i). Preprocessing time is $O($ $)$
(ii). Storage space (memory usage) is $O($ $)$
(iii). Query time is $O($ $)$

(4). [9 points] Suppose I claim to have an algorithm, ALG, which computes the convex hull of n points in 4D, and does so in the best possible worst-case time (in terms of n). How fast is ALG (in terms of n)? Give your answer in terms of big-Oh notation.

Explain briefly how you could use ALG to compute the Delaunay triangulation ("tetrahedralization") of n points in 3D.

(5). [12 points] We want to solve the following query problem: Given a set S of n disjoint line segments in the plane, determine how many segments are stabbed by a query line ℓ.

(a). Describe briefly a data structure and a method for this problem. Try to be as efficient as possible, especially in query time.

(i). Preprocessing time is $O($ $)$
(ii). Storage space (memory usage) is $O($ $)$
(iii). Query time is $O($ $)$

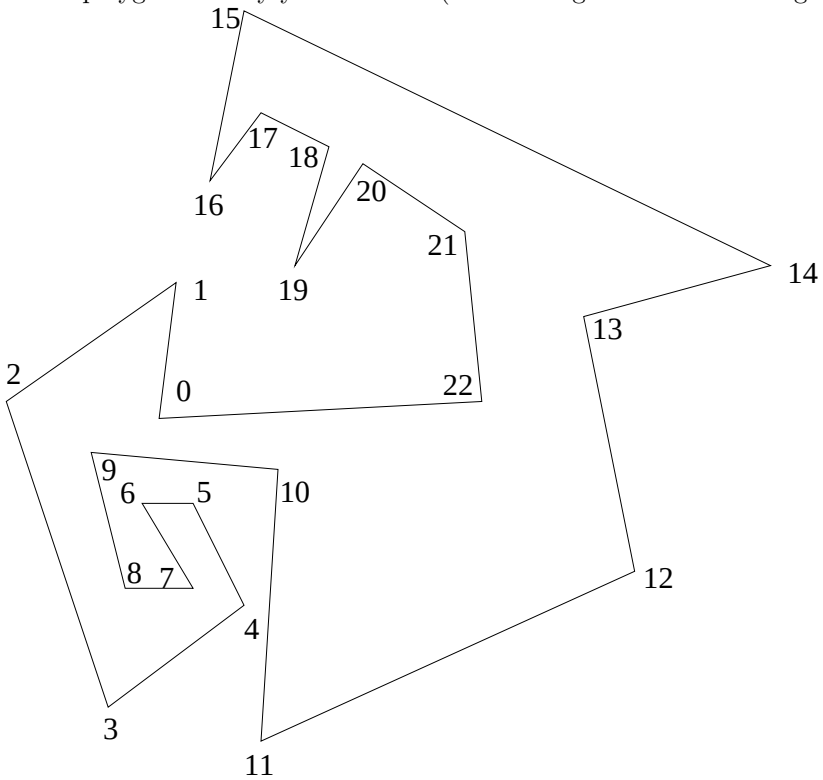
(b). Answer the question now assuming that the query line is known to be horizontal or vertical.

(i). Preprocessing time is $O($ $)$
(ii). Storage space (memory usage) is $O($ $)$
(iii). Query time is $O($ $)$

(6). [9 points] Let $S = \{p_1, \ldots, p_n\}$ be a set of n points in the plane. Describe briefly how you would preprocess S into a data structure that will support very efficient queries of the form: For query point q, report all points of S that are within L_1 distance d of q. (Recall that the L_1 distance between p and q is given by $d_1(p,q) = |p_x - q_x| + |p_y - q_y|$.)

(7). [9 points] Let $L = \{\ell_1, \ldots, \ell_n\}$ be a set of n lines in the plane. Describe briefly how you would preprocess L into a data structure that will support queries of the following form: For query line ℓ, report the points where ℓ crosses the lines ℓ_i, and give these points in order of increasing x-coordinate.

(8). [9 points] (a). By inspection, obtain the *point* guard number for P, allowing guards to be placed at *any* point (interior or boundary) of the polygon. Justify your answer! (Give an argument that fewer guards cannot suffice.)



(b). How many (vertex) guards does the Art Gallery Theorem guarantee are sufficient for this polygon?