AMS 345/CSE 355 Joe Mitchell

COMPUTATIONAL GEOMETRY
Practice Final

Closed book, closed notes. You have 75 minutes to complete the exam.

(1). [15 points] How efficiently can the convex hull of n points in 3D be computed? Give your answer in terms of big-Oh notation, with a brief justification showing that you know roughly how it is obtained.

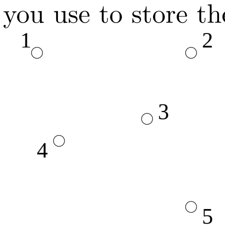
(2). [20 points] For the set S of 5 points shown below, do the following:

(a). Sketch the Voronoi diagram. How many Voronoi vertices are there? Voronoi edges? Voronoi regions?

(b). Sketch the Delaunay diagram. How many Delaunay edges are there? How many Delaunay faces?

(c). Draw the directed nearest neighbor graph.

(d). What type of data structure would you use to store the Voronoi diagram?



(3). [20 points] For each of the computations below indicate how efficiently one can perform the calculation, in terms of $O(...)$ notation (e.g., $O(n)$, $O(\log n)$, $O(n^2)$, $O(n \log n)$). Try to give the best (lowest) upper bound possible. Justify your answer very briefly.

(a). Determine whether or not a set S of n line segments in the plane has *any* point of intersection among the segments.

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

(c). Given a winged edge data structure for a Delaunay diagram of n points in the plane, compute the convex hull of the points.

(d). Preprocess a simple polygon P, having n vertices (given in counterclockwise order), in order to support efficient queries of the form: Is point q inside P? State the complexity of both the preprocessing and the query.

(e). Preprocess a set S of n points in the plane to support efficient queries of the form: Does the unit circle centered at point q contain any points of S?

(4). [20 points] For each of the following statements, state whether it is **ALWAYS TRUE**, **SOMETIMES TRUE** (but sometimes false), or **NEVER TRUE**.

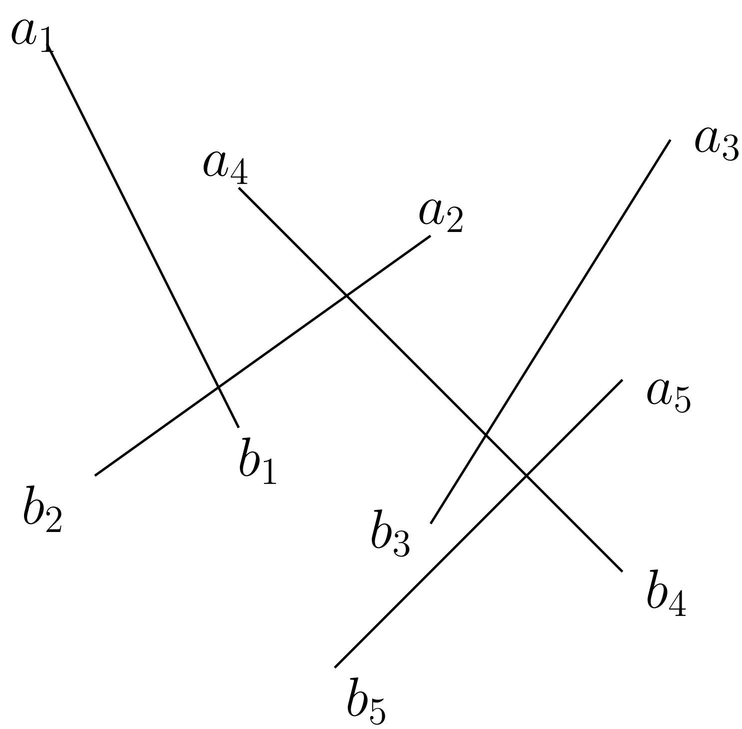
(a). The nearest neighbor graph of n points in the plane is connected.

(b). The Delaunay diagram of n points in the plane is connected.

(c). An optimal travelling salesperson tour on n points in the plane is the boundary of a simple polygon.

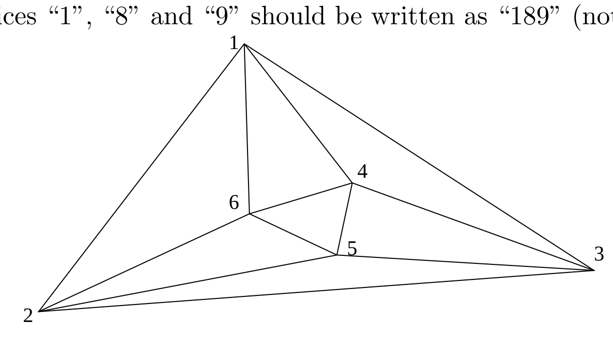
(d). A polygonal subdivision of the plane into n convex polygonal cells is the Voronoi diagram for some appropriately chosen set of sites, S, with one point per cell.

(5). [10 points] When the Bentley-Ottmann sweep algorithm is performed on the set of line segments below, in order to report all points of intersection among them, give the priority queue of events and the sweep line status just *after* the event that the sweep line hits point a_2. (Assume a horizontal sweep line, from top to bottom.)



(6). [15 points] Build the Kirkpatrick point location hierarchy for the triangulation shown below. At each step, when you identify an independent set, apply Algorithm 7.4 on page 277, breaking ties when you select a node in favor of the lowest numbered vertex. When you retriangulate a hole, use the simple ear-clipping algorithm (`Triangulate`, page 39), starting at the (rightmost) bottommost vertex of the hole (as "$v0$" in `Triangulate`, the first one tested for earity), and proceeding counterclockwise.

Draw the final hierarchy as a DAG, with each node of the hierarchy labeled by the triangle to which it corresponds. (When you label a node, please list the triangle as a triple with the vertex indices in order; e.g., the triangle with vertices "1", "8" and "9" should be written as "189" (not as "819" or "918", etc).)



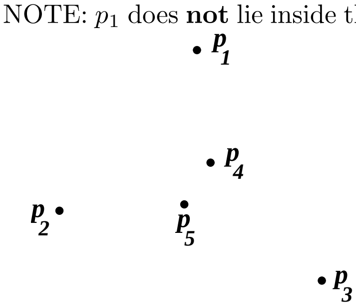
AMS 345/CSE 355 Joe Mitchell

COMPUTATIONAL GEOMETRY
Another Practice Final

Closed book, closed notes. You have 75 minutes to complete the exam.

(1). [21 points] For the set S of 5 points shown below, do the following:

(a). Draw the Delaunay diagram. NOTE: p_1 does **not** lie inside the circle(p_2, p_5, p_4); also, $|p_3p_4| > |p_3p_5|$.



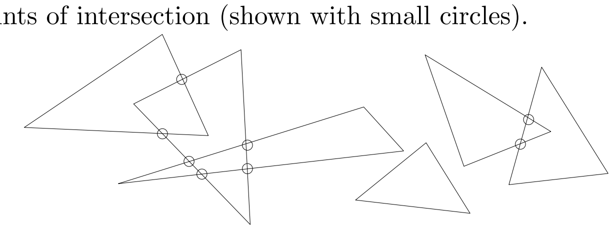
How many Delaunay edges are there? How many Delaunay faces?

(b). Sketch the Voronoi diagram. How many Voronoi vertices are there? How many Voronoi edges are there? How many Voronoi regions are there?

(c). Draw the Relative Neighborhood Graph (RNG). (Recall from HW6 that the RNG joins p_i and p_j with an arc iff $|p_i - p_j| \leq \max_{m \neq i,j}\{|p_i - p_m|, |p_j - p_m|\}$.)

(2). [30 points] For each of the computations below indicate how efficiently one can perform the calculation, in terms of $O(...)$ notation (e.g., $O(n)$, $O(\log n)$, $O(n^2)$, $O(n \log n)$). Try to give the best (lowest) upper bound possible.

(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. For example, in the figure below, we would need to report 8 points of intersection (shown with small circles).



(b). Compute the nearest neighbor graph (NNG) of n points in the plane.

(c). Given a winged-edge data structure for a Delaunay diagram of n points in the plane, compute the winged-edge data structure of the Voronoi diagram of the same n points.

(d). Given n red points, n blue points, and n green points in the plane, compute a data structure that allows efficient answers to queries of the form: Which color point is closest to q? State the preprocessing time, storage space, and query time.

(e). Given a set $\mathcal{L}$ of n lines, build a winged-edge data structure for their arrangement, $A(\mathcal{L})$.

(f). Given n points in 4D, compute their convex hull.

(3). [24 points] For each of the following statements, state whether it is **ALWAYS TRUE**, **SOMETIMES TRUE** (but sometimes false), or **NEVER TRUE**. Briefly justify your answer.

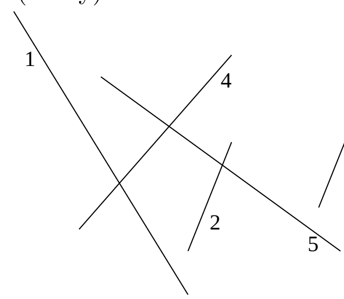
(a). Given n red points, n blue points, and n green points in the plane, there exists a line L that simultaneously splits the red, blue, and green points in half ($n/2$ on each side of L). (You may assume that n is even.)

(b). For any set of n distinct points in the plane, the Voronoi edges and Voronoi vertices form a connected graph.

(c). If (p, q) is a Delaunay edge, then the line segment pq crosses the Voronoi edge that is shared by the cells $V(p)$ and $V(q)$.

(d). If $\{p_1, \ldots, p_k\}$ is a set of distinct collinear points in the plane, then the duals of these k points are parallel to each other (using the standard definition of duality from the text).

(4). [10 points] When the Bentley-Ottmann sweep algorithm is performed on the set of line segments below, in order to report all points of intersection among them, give the priority queue of events and the sweep line status just *after* the event that the sweep line hits point a_2. (Assume a horizontal sweep line, from top to bottom.) Assume that we use the notation that each segment $s_i = (a_i, b_i)$ has upper endpoint a_i and lower endpoint b_i, and x_{ij} denotes the point (if any) at the intersection of segment s_i and segment s_j.



In the notation we have used in class and on the homeworks, I am asking you to fill in the row of the table below that starts with "a_2". (You may fill in more of the table too, if it helps you.)

Event	Event Queue, Q	Sweep Status, $\mathcal{L}$
-		
a_2		

(5). [15 points] Build the Kirkpatrick point location hierarchy for the triangulation shown below. At each step, when you identify an independent set, apply Algorithm 7.4 on page 277, breaking ties when you select a node in favor of the lowest numbered vertex. When you retriangulate a hole, use the simple ear-clipping algorithm (`Triangulate`, page 39), starting at the bottommost vertex of the hole (as "$v0$" in `Triangulate`, the first one tested for earity), and proceeding counterclockwise.

Draw the resulting portion of the hierarchy as a DAG, with each node of the hierarchy labeled by the triangle to which it corresponds. (When you label a node, please list the triangle as a triple with the vertex indices in order; e.g., the triangle with vertices "1", "8" and "9" should be written as "189" (not as "819" or "918", etc).)

I include some copies of the point set, in order to assist you in your drawings.