ICS 164 - Spring 2019 - Final exam

Name:

Student number:

UCInet ID:

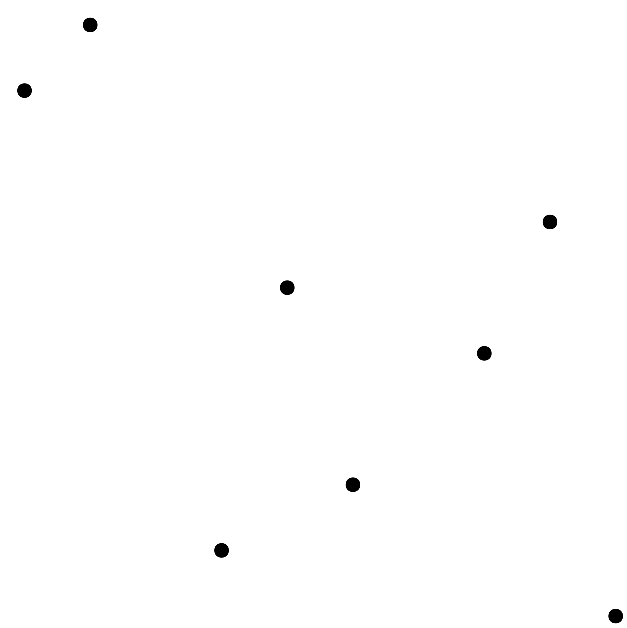
This is a closed book, closed note exam.

Do not open this exam until told to start by the instructor.

Please write your answers ONLY on the front side of each page.
Answers written elsewhere will be discarded and not graded.

You may use the back sides of the pages as scratch paper.
Do not unstaple the pages.

1. (15 points) Draw a kD-tree on the following set of points. (Start with a vertical splitting line. When the construction uses the median of an even number of values, choose the smaller value of the two middle elements, the one closer to the bottom left of the figure. Just draw the separating lines; do not separately draw the tree structure.)

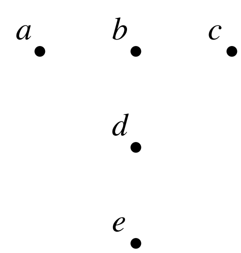


2. (15 points) What shape is the Minkowski sum of two squares of side length one, both of which are parallel to the coordinate axes?

3. (15 points) For the set of five points depicted below,

   (a) Which points are vertices of the convex hull?

   (b) Which points have Voronoi cells that extend to infinity?



4. (15 points) Suppose that a set of n points is in general position (no three in a line, no four on a circle) and has k vertices on its convex hull. How many triangles does its Delaunay triangulation have, as a function of n and k?

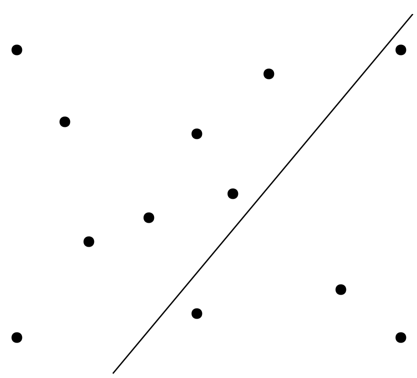
   (Do not use O-notation. Hint: Recall that the Delaunay triangulation can be constructed by repeatedly adding points and then flipping pairs of triangles. Start by adding all the convex hull vertices before adding any other points or flipping anything. How does each of these operations change the total number of triangles?)

5. (15 points) Suppose that we are given a collection of n intervals $[\ell_i, r_i]$ (where ℓ_i is the left endpoint and r_i is the right endpoint of the ith interval). We wish to construct an interval tree with as few tree nodes as possible for these intervals. Describe how to test in linear time, from this information, whether it is possible to construct an interval tree that has only a single node. (You may assume that $\ell_i < r_i$ for each i, and that each interval contains its two endpoints. Your algorithm should not actually construct the interval tree, only determine whether it exists.)

6. (15 points) Find a set of non-crossing line segments in the plane, in general position (meaning that no two segment endpoints have the same x-coordinate) so that in their trapezoidal decomposition one of the trapezoids t has only one neighboring trapezoid to its left (across its left vertical edge) and only one neighboring trapezoid to its right (across its right vertical edge). Draw the whole trapezoidal decomposition for your set of segments, and label trapezoid t in it.

7. (15 points) The plane sweep algorithm for finding all crossing points of an arrangement of line segments uses two data structures, a binary search tree and a priority queue. Suppose that we use the version of the algorithm described in the lecture, in which the sweep is left-to-right, and the priority queue is ordered by the x-coordinates of the endpoints and crossing points. What elements are stored in the binary search tree, and how are they ordered?

8. (15 points) Suppose we are performing the range query depicted for the point set below, in which we want to list all of the points below the line shown.



   (a) In the version of the range reporting data structure for this problem that does not use fractional cascading, how many binary searches will it perform?

   (b) How many of the given data points will it test for being above or below the line?