1. Please define each of the following terms:
   (a) line arrangement
   (b) convex hull
   (c) $\epsilon$-net
   (d) upper envelope
   (e) Delaunay triangulation

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

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

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

3. Suppose you are given a trapezoidal decomposition of a simple polygon $P$. Briefly describe a linear-time method for producing a triangulation of $P$.

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

5. Prove that the total size of the arrangement of $n$ planes in $\mathbb{R}^3$ is $O(n^3)$.

6. Recall that an event holds with $n$-exponential probability if the probability for this event is at least $1 - 1/b^{nc}$ for some constants $b > 1$ and $c > 0$. Let $S$ be a collection of $n$ circles in the plane, and let $R$ be a $\sqrt{n}$-sized random sample of $S$. Suppose $C$ is a circle that intersects $n/4$ circles in $S$. Using one of the well-known Chernoff bounds, show that $C$ intersects at least one circle of $R$ with $n$-exponential probability.
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 $C$ be a unit cube in $\mathbb{R}^3$ and let $S$ be a set of $n$ points inside $C$. Describe an efficient algorithm for finding the largest empty sphere with its center inside $C$. What is the running time of your method?

8. Let $S$ be a set of $n$ points in the plane (unsorted). Describe a linear-time method for determining the edges of the convex of $S$ that are intersected by a given vertical line $L$. (Hint: consider the dual problem.)

9. Suppose you are given two convex polygons $P$ and $Q$ that are separated by a line (but you don’t know which one). Give an efficient algorithm for finding the closest pair of points $(p, q)$ such that $p$ is on the boundary of $P$ and $q$ is on the boundary of $Q$, assuming that $P$ and $Q$ are already stored in main memory in arrays. What is the running time for your method?

10. Suppose you are given a set $S$ of $n$ points in the plane. Define the flatness of $S$ to be the minimum distance between two parallel lines containing the points of $S$ between them.

   (a) Describe an efficient algorithm for finding the flatness of $S$. What is the running time of your method?

   (b) Generalize this definition of flatness to 3-dimensional sets of points.

   (c) Speculate on how you might extend your 2-dimensional approach to design an algorithm for determining the flatness of a 3-dimensional set of points.