You can skip one of the first five problems, but problem 6 is required.

1. (a) Prove that any simplex is collapsible.

   (b) For any simplicial complex Δ, let $f_i(\Delta)$ denote the number of simplices in Δ with $i+1$ vertices (that is, with dimension i). In particular, $f_{-1}(\Delta) = 1$ for any complex Δ. The *Euler characteristic* of Δ, denoted $\chi(\Delta)$, is defined as the following alternating sum:

   $$\chi(\Delta) = \sum_i (-1)^i f_i(\Delta).$$

   Prove that every collapsible simplicial complex has Euler characteristic zero.

   (c) Describe a non-collapsible simplicial complex with Euler characteristic zero.

2. (This problem appeared on this semester's theory qual.) A *swap* transforms one permutation into another by exchanging one pair of adjacent elements. The *swap distance* between two permutations is the minimum number of swaps required to transform one into the other. For example, we can transform $\langle 1,4,3,2\rangle$ into $\langle 1,2,3,4\rangle$ with three swaps

   $$\langle 1,4,\underline{3,2}\rangle \to \langle 1,\underline{4,2},3\rangle \to \langle 1,2,\underline{4,3}\rangle \to \langle 1,2,3,4\rangle$$

   but not with two swaps, so the swap distance between those two permutations is 3.

   (a) Prove that the expected swap distance between two random permutations of $\{1,2,\ldots,n\}$ is $\Theta(n^2)$.

   (b) Describe an efficient algorithm to compute the swap distance between two given permutations of $\{1,2,\ldots,n\}$.

   (c) Prove that your algorithm is optimal in some appropriate model of computation.

3. The MaxGap problem asks, given a set $X = \{x_1, x_2, \ldots, x_n\}$ of real numbers, to find the pair of adjacent elements that is furthest apart. That is, we want to find two elements x_i and x_j such that no element of X lies strictly between x_i and x_j, and the distance $|x_i - x_j|$ is maximized.

   (a) Prove an $\Omega(n \log n)$ lower bound for MaxGap in the algebraic computation tree model. *[Hint: Use a reduction from another problem.]*

   (b) Describe an algorithm to solve the MaxGap problem in $O(n)$ time. *[Hint: Do something that the algebraic computation tree model doesn't allow.]*

4. Consider the following version of the traveling salesman problem: Given $\binom{n}{2}$ real numbers, representing weights on the edges of a complete n-vertex graph, and another real number L, is there a Hamiltonian cycle of length at most L?

   (a) Show that any algebraic decision tree that solves the traveling salesman problem has depth $\Omega(n^2)$.

   (b) Describe a linear decision tree of polynomial depth that solves the traveling salesman problem.

5. Let X be a set of n points in the plane. Recall that the *convex hull* of X is the smallest convex set containing X; intuitively, we can define this by wrapping a rubber band around the points. A point in X is called *extreme* if is it a vertex of the convex hull of X.

   Suppose we wanted to compute the extreme points in an n-point set using a "ruler and compass" algorithm.. Such an algorithm maintains a set of points, lines, and circles, subject to the following operations:

   - Construct a directed line between any two points.
   - Construct the circle with one point at the center and another point on the boundary.
   - Construct a point at the intersection of two lines (if they intersect).
   - Construct the points at the intersection of a line and a circle (if they intersect).
   - Construct the points at the intersection of two circles (if they intersect).
   - Test whether a point is on the left side, on the right side, or on a directed line.
   - Test whether a point is inside, outside, or on the boundary of a circle.

   Prove that any "ruler and compass" algorithm needs $\Omega(n \log n)$ steps to compute the extreme points in a set of n points in the plane. *[Hint: Formalize the model of computation.]*

6. This problem asks you to derive lower bound for linear decision trees based on counting the number of faces of a convex polyhedron. Recall that a *convex polyhedron* is the intersection of a finite number of closed linear halfspaces. A hyperplane *supports* a polyhedron if it does not intersect the interior of the polyhedron. A *proper face* of a polyhedron Π is the intersection of (the boundary of) Π with any supporting hyperplane; proper faces are lower-dimensional polyhedra. A *face* is either a proper face or the entire polyhedron. A *facet* is an face with co-dimension 1.

   For example, the cube is a polyhedron defined by intersecting six halfspaces:

   $$x \geq 0, \qquad x \leq 1, \qquad y \geq 0, \qquad y \leq 1, \qquad z \geq 0, \qquad z \leq 1.$$

   This polyhedron has a total of 28 faces: the entire cube, six (2-dimensional) square facets, twelve (1-dimensional) edges, eight (0-dimensional) vertices, and the (-1-dimensional) empty set. In general, the number of faces is (very crudely!) limited by the following theorem:

   The Upper Bound Theorem. *The intersection of n halfspaces in $\mathbb{R}^d$ has $O(n^{\lfloor d/2 \rfloor})$ faces.*

   (a) Prove that if a convex polyhedron Π can be written as the union of two convex polyhedra $\Pi_1 \cup \Pi_2$, then $F(\Pi) \leq F(\Pi_1) + F(\Pi_2)$.

   (b) Let Π be a convex polyhedron in $\mathbb{R}^n$, and let $F(\Pi)$ denote the number of faces of Π. Prove that any linear decision tree that decides whether an input point $x \in \mathbb{R}^n$ is inside Π has depth $\Omega(\log F(\Pi))$.

   (c) Now let Π be a fixed convex polyhedron with f faces in $\mathbb{R}^d$. Prove a lower bound for the following problem: Given n points in $\mathbb{R}^d$, do they all lie inside Π? *[Hint: The problem really asks whether a single point is inside a polyhedron $\hat{\Pi}$ in $\mathbb{R}^{nd}$; what are the faces of this polyhedron?]*