Typeset your homework solutions in LaTeX and email a PostScript or PDF file (preferably as a MIME attachment) to jeffe@cs.uiuc.edu, with "CS 497" in the subject. I'll also grudgingly accept printed solutions with hand-drawn pictures. Groups of up to three people can turn in a common solution. Since this is a graduate class, I *expect* you to consult textbooks, journals, conference proceedings, the web, and each other, but you must cite your sources.

A *star on any problem means that I don't know the solution (yet (?)). Starred problems may be impossible, or they may be completely trivial. Good luck!

1. A simple polygon is a circular sequence of line segments joined end-to-end with no other intersections.

   (a) The "three-penny algorithm" from Graham's scan computes the convex hull of some simple polygons, but not all. Describe a simple polygon P for which the three-penny algorithm fails to compute the convex hull.

   (b) Describe an algorithm to construct the convex hull of any simple n-vertex polygon in $O(n)$ time. *[Hint: This is harder than it looks!]*

2. (a) Describe an algorithm to compute the intersection of two convex polygons in $O(n)$ time, where n is the total number of edges in both polygons.

   (b) Describe an algorithm to compute the convex hull of two convex polygons in $O(n)$ time, where n is the total number of edges in both polygons. This immediately implies an $O(n \log n)$ "mergehull" algorithm! *[Hint: Use part (a).]*

3. Let P be a set of n points in the plane. A point $p \in P$ is *maximal* (or *Pareto-optimal*) if no other point in P is both above and to the right of p.

   (a) Show that not every extreme point of P is maximal, and not every maximal point is on the convex hull of P. *[Hint: There is a four-point solution.]*

   (b) Describe an algorithm to compute the m maximal points in P in $O(n \log m)$ time. *[Hint: Despite part (a), modify a convex hull algorithm.]*

4. (a) Let V be a set of vectors in $\mathbb{R}^3$. The *acute hull* of V is the set of vectors in $\mathbb{R}^3$ that form an acute angle with every vector in V. Describe an algorithm to compute the acute hull of a set of n vectors in $O(n \log n)$ time. *[Hint: This is a familiar problem in disguise.]*

   (b) Let G be a set of *good* points and let B be a set of *bad* points in the plane. A point p is *acceptable* if it is closer to *every* good point and farther from *every* bad point than the origin $(0, 0)$. Describe an algorithm to compute the set of acceptable points in $O(n \log n)$ time, given G and B as input. *[Hint: So is this.]*

*5. Describe an algorithm that finds, given a set of n points in the plane, the median vertex of its upper convex hull, in $O(n)$ time. A randomized algorithm that *usually* finds an *approximate* median is fine. Such an algorithm would immediately imply a variant of quickhull that runs in $O(n \log h)$ time without prune and search.