General Instructions: Please typeset your homework using LaTeX, with each problem starting on a new page. See the course web site for advice on producing and importing figures. *Stars indicate problems that I don't already know how to solve; these problems may or may not be open.

I *strongly* encourage you to work together, but each student must turn in their own solutions. Similarly, you may use any resource at your disposal—human, printed, or electronic—but you must not copy anything verbatim, and you must cite *every* source you use, including the other students you work with. In other words, follow the same standards of scholarship you would use for a research paper.

---

1. (a) Describe an algorithm to compute the intersection of two convex n-gons in $O(n)$ time.

   (b) Describe an algorithm to construct the *convex hull* of two convex n-gons in $O(n)$ time. *[Hint: Use part (a).]*

   (c) Describe and analyze 'merge-hull'!

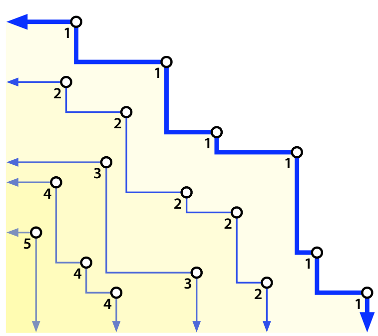
2. A *simple* polygon is a circular sequence of line segments joined end-to-end with no other intersections.

   (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) Prove that Graham's scan (either the version described in class or the version described in the lecture notes) computes the convex hull of any set of points!

   (c) 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!]*

3. Let P be a set of n points in the plane. A point p in P is *Pareto-optimal* if no other point in P is both above and to the right of p. The sorted sequence of Pareto-optimal points describes a *staircase* with all the points in P below and to the left. The *staircase layers* of P are defined recursively as follows. The first staircase layer is just the staircase; for all $k > 1$, the kth staircase layer is the staircase of P after the points in the first $k - 1$ staircase layers have been deleted.



A set of points with five staircase layers.

   (a) Describe an algorithm to compute the staircase of P in $O(n \log p)$ time, where p is the number of Pareto-optimal points. *[Hint: There are at least two different ways to do this.]*

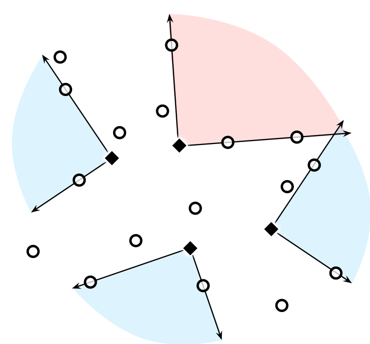
   (b) Describe and analyze an algorithm to compute the staircase layers of P in $O(n \log n)$ time. Your algorithm should label each point with an integer indicating which staircase layer contains it, as shown in the figure above.

   Assume that no two points in P have the same x- or y-coordinates.

4. Let P be a set of points in the plane. A *wedge* is a region of the plane bounded by two rays from a common point, called the *apex* of the wedge. A *vantage point* is the apex of a wedge satisfying three conditions:

   - The angle between the bounding rays is exactly 90 degrees.
   - Each bounding ray passes through at least one point in P.
   - No point in P lies in the interior of the wedge.

   A vantage point is *extreme* if one of the bounding rays of its wedge passes through at least two points of P. Finally, the *vantage set* of P is the set of all vantage points.



Four vantage points (black diamonds) for a set of points (white circles).
The highest vantage point (looking northeast) is extreme.

   (a) Prove that the vantage set of P is the union of a finite number of closed circular arcs. What are the centers and endpoints of these arcs? *[Hint: Consider the case $n = 2$.]*

   (b) Prove that there are at most $O(n)$ extreme vantage points. *[Hint: Charge each extreme vantage point to one or more points in P, so that each point in P is charged $O(1)$ times.]*

   (c) Prove that the vantage set of P is the union of at most $O(n)$ closed circular arcs. *[Hint: This should follow easily from parts (a) and (b).]*

   (d) Describe and analyze an algorithm to compute the vantage set of P. Your algorithm should return the set of $O(n)$ circular arcs described in part (c). To keep your algorithm as simple as possible, assume the points P are in general position. For full credit, your algorithm should run in near-quadratic time. *[Hint: Consider a rotating staircase.]*

   In particular, your solution should describe the underlying primitives used by your algorithm, both geometrically (as a relationship among a small subset of points) and algebraically (as the sign of a multivariate polynomial), as well as the corresponding 'general position' assumptions. For example:

   LEFTORRIGHT(p, q):
   Geometric: Is point p left or right of point q?
   Algebraic: $\text{sgn}(p_x - q_x)$
   General position: No two points lie on a vertical line.

   *(e) Describe and analyze an algorithm to compute the vantage set of P in $O(n \log n)$ time, or prove that no such algorithm is possible. *[Hint: I know how to do this in $O(n \operatorname{polylog} n)$ time, but it isn't pretty.]*

5. (a) Suppose we are given the Voronoi diagram of a set P of n points in the plane, but we are not given the set P itself. Describe an algorithm to reconstruct the point set P from its Voronoi diagram in $O(n)$ time. Assume that every Voronoi vertex has degree 3. If there is more than one point set consistent with the given diagram, just return one such set.

   *(b) Remove the simplifying assumptions from the previous problem. Describe an efficient algorithm to compute a concise description of *all* point sets P consistent with the given Voronoi diagram. *Do not* assume that every Voronoi vertex has degree 3.

   *(c) Now suppose we are given only the *vertices* of a Voronoi diagram, but not the Voronoi edges or the sites. Describe an efficient algorithm to reconstruct a set P of points (or even better, a description of all such sets) whose Voronoi diagram has the given set of vertices, or prove that no such algorithm is possible. If necessary, assume that every Voronoi vertex has degree 3 in the (unknown) Voronoi diagram.