CS 498 TC ✧ Computational Geometry ✧ Spring 2026

Homework 3

Due Wednesday, May 6, 2026 at 8pm

---

1. Let P be a set of points in the plane in general position. For any line ℓ, let $depth(\ell, P) = \min\{|P \cap \ell^+|, |P \cap \ell^-|\}$, where ℓ^+ and ℓ^- are the closed halfplanes bounded by ℓ. For any point q in the plane (not necessarily in P), let $depth(q, P) = \min\{depth(\ell, P) \mid q \in \ell\}$. Finally, the ***Tukey depth*** of P is $\max\{depth(q, P) \mid q \in \mathbb{R}^2\}$.

   (a) **Warmup, not for submission:** Prove that $depth(q, P) > 0$ if and only if q lies in the convex hull of P.

   (b) Describe and analyze an algorithm that computes $depth(q, P)$, given the point q and point set P as input.

   (c) Describe and analyze an algorithm that determines whether there is a point q such that $depth(q, P) \geq k$, given the set P and the integer k as input.

   This problem can be solved in $O(n^2)$ time using only techniques introduced in class, but slower correct algorithms are worth significant partial credit. This is *not* the fastest algorithm known.

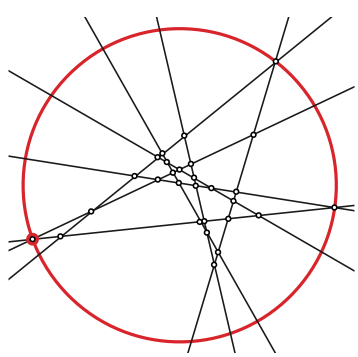
   [Hint: What is the natural dual interpretation of depth(q, P)?]

2. Suppose you are given a set L of n lines in the plane in general position.

   (a) Describe and analyze an algorithm to compute the leftmost vertex of the arrangement of L.

   (b) Describe and analyze an algorithm to compute the smallest circle containing every vertex of the arrangement of L.

   Both of these problems can be solved in $O(n^2)$ expected time by running Welzl's MiniDisk algorithm on the $\binom{n}{2}$ intersection points, but you can do better. *[Hint: Convex hull.]*



3. In class we saw the classical *funnel* algorithm to compute shortest paths inside a triangulated simple polygons. How would you modify this algorithm to find shortest paths in a polygon with holes?

   (a) Describe and analyze an algorithm to compute the shortest path between two given points in the interior of a polygon with *one* hole. *[Hint: Which way does the path go around the hole?]*

   (b) Describe and analyze an algorithm to compute the shortest path between two given points in the interior of a polygon with *two* holes.

   (c) **Extra credit:** Describe and analyze an algorithm to compute shortest paths in a polygon with h holes; analyze your algorithm as a function of both n (the total number of vertices) and h (the number of holes).

   In all cases, you can assume that you are given a triangulation of the input polygon. For full credit, your algorithms should run in $O(n)$ time (for any constant h).