Due date: October 7, 2025

Problem 1: (i) Describe and analyze an algorithm that, given a set $P \subset \mathbb{R}^2$ of n points, computes in $O(n \log n)$ time the line ℓ^* that minimizes the maximum distance from a point of P, i.e. $\ell^* = \arg\min_\ell \max_{p \in P} d(p, \ell)$ where $d(p, \ell) = \min_{x \in \ell} ||p - x||_2$. (**Hint:** *Consider the two lines parallel to ℓ^* at distance $\max_{p \in P} d(p, \ell)$. What properties do these lines have?*)

(ii) Describe an $O(n \log n)$-time algorithm to compute the minimum-area rectangle containing P, by modifying your solution to prt (i).

Problem 2: Let $\triangle$ be a triangle in $\mathbb{R}^2$. For a point $x \in \mathbb{R}^2$, $\triangle + x$ is a *translate* of $\triangle$ (if a, b, c are vertices of $\triangle$, then the vertices of $\triangle + x$ are $a + x, b + x, c + x$.) Let S be a set of n translates of Δ. Let $U(S) \subseteq \mathbb{R}^2$ denote the union of triangles in S, i.e., the set of points that lie in at least one triangle of S. Each connected component of $U(S)$ is a polygon (with possible holes). The total number of vertices on $U(S)$ is known to be $O(n)$. Describe an $O(n \log^2 n)$-time algorithm to compute the boundary of the union of $\mathcal{S}$. (You can assume that once you have all the edges of a planar subdivision Π, you can compute a suitable representation of Π in linear time.) (**Hint:** *Describe a divide-and-conquer algorithm.*)

Problem 3: Let R be a set of pairwise-disjoint "red" segments in $\mathbb{R}^2$, and let B be a set of pairwise-disjoint "blue" segments; set $n = |R| + |B|$. Describe an $O(n \log^2 n)$ time algorithm to count the number of red-blue intersections. Show that the run-time can be improved to $O(n \log n)$ using fractional cascading. (**Hint:** *Use segment tree on the x-projections of $R \cup B$.*)

Problem 4:

(a) Let P be a convex polygon in $\mathbb{R}^2$ with n vertices. Assuming that the sequence of vertices of P are stored in an array or a red-black tree, show that the tangents of P from a point lying outside P can be computed in $O(\log n)$ time.

(b) Let S be a set of n points in $\mathbb{R}^2$. Show that each step of the Jarvis march algorithm described in class can be implemented in $O((n/k) \log k)$ time, where k is the number of vertices in the convex hull of S. You can assume that the value of k is known. (**Hint:** *Partition S into $\lceil \frac{n}{k} \rceil$ sets, each of size at most k. Use (a) to compute the next vertex of the convex hull in a total of $O((n/k) \log k)$ time.*)

Problem 5: Let $P, w : P \to \mathbb{R}_{\geq 0}$ be a weighted point set in $\mathbb{R}^2$. Define the *power distance* of a point $x \in \mathbb{R}^2$ to a point $p \in P$ to be

$$\mathsf{d}(p, x) = \sqrt{\|p - x\|^2 - w_p^2}.$$

Prove that the Voronoi cell of a point in P under the power distance is a convex polygon. Describe an $O(n \log n)$ expected time algorithm to compute the Voronoi diagram of P under the power distance, which is referred to as the *power diagram* of P.