Due date: November 13, 2025

Problem 1: Let S be a set of n points in $\mathbb{R}^2$. A point $x \in \mathbb{R}^2$ is called a *center point* if any half-plane containing x contains at least $\lfloor n/3 \rfloor$ points of S. It is known that a center point always exists.

(i) Describe an $O(n^2)$ time algorithm to compute a center point of S.

(ii) Describe an $O(n)$ time randomized algorithm to compute an approximate center point $\tilde{x}$ of S, i.e., any half-plane containing $\tilde{x}$ contains at least $n/4$ points of S.

Problem 2: Let $S = \{\ell_1, \ldots, \ell_n\}$ be a set of n line segments in $\mathbb{R}^2$. Describe an algorithm for preprocessing S into a data structure that for a query line ℓ counts the number of segments of S intersected by ℓ.

(i) Describe a data structure with $O(n^2)$ storage and preprocessing time, and $O(\log n)$ query time. (**Hint:** *Use duality*)

(ii) Describe a data structure with $O(n \log n)$ storage and $O(n^{\frac{1}{2}+\varepsilon})$ query time.

(**Hint:** *Construct a 2-level partition tree.*)

Problem 3: Let S be a set of n points in $\mathbb{R}^3$. A cylinder $C = C(r, \ell)$ of radius r with axis ℓ is the set of points at distance r from ℓ. The distance of a point $p \in \mathbb{R}^3$ from C is $\mathrm{dist}(p, C) = |d(p, \ell) - r|$, where $d(p, \ell) = \min_{x \in \ell} \|p - x\|$. The *best-fit cylinder of S* is the cylinder C^* that minimizes the maximum distance between C^* and S, i.e.,

$$C^* = \arg\min_{C} \max_{p \in S} \mathrm{dist}(p, C).$$

For a given parameter $\varepsilon > 0$, describe an $O(n + \varepsilon^{-O(1)})$ time ε-approximation algorithm to compute the best-fit cylinder, i.e., a cylinder $\widetilde{C}$ such that

$$\max_{p \in S} \mathrm{dist}(p, \widetilde{C}) \leq (1 + \varepsilon) \cdot \max_{p \in S} \mathrm{dist}(p, C^*).$$

(**Hint:** *Show that a core-set of size $\varepsilon^{-O(1)}$ exists for this problem.*)

Problem 4: Let S be a set of points in $\mathbb{R}^d$. Show that the randomized LP algorithm discussed in the class can be extended to compute the smallest ball containing S in $O(n)$ expected time.