CMSC 754: Final Exam

This exam is closed-book and closed-notes. You may use two sheets of notes (front and back). Write all answers on the exam paper. If you have a question, either raise your hand or come to the front of class. Total point value is 135 points. Good luck!

In all problems, unless otherwise stated, you may assume that inputs are in *general position*. You may make use of any results presented in class and any well known facts from algorithms or data structures. If you are asked for an $O(T(n))$ time algorithm, you may give a *randomized* algorithm with *expected* time $O(T(n))$.

Problem 1. (35 points) Short-answer questions. Unless requested, explanations are not required.

(a) (5 points) Given a set of points P in $\mathbb{R}^2$, there are k_1 edges on P's upper hull and k_2 edges on P's lower hull. What can we say about the number of vertices on the upper and lower envelopes of the dual line arrangement P^*? (Assume the standard dual transformation we use in class.)

(b) (15 points) For each of the following, indicate whether it *can or cannot* be solved by a reduction to a constant number of instances of linear programming of size $O(n)$ in a space of constant dimension. If it can be, indicate the *dimension* of the LP instance. *Do not* give the LP formulation.

(i) Given a set of n vertical line segments in $\mathbb{R}^3$ (that is, all parallel to the z-axis), does there exist a plane ℓ that intersects all of these segments?

(ii) Given a set of n vertical line segments in $\mathbb{R}^2$, does there exist a line ℓ that stabs at least half of these segments?

(iii) Given a set of n points in $\mathbb{R}^2$, compute the slab (region bounded by two parallel lines) of minimum vertical height that encloses all these points.

(iv) Given a set of n points in $\mathbb{R}^2$, compute the closest pair of points.

(c) (5 points) Given an n-element point set P in $\mathbb{R}^2$, we want to compute the largest circular disk that has its center in P's convex hull (or on its boundary) and contains no points of P in its interior. Describe a set of $O(n)$ points, computable in $O(n \log n)$ time, such that the optimal center lies at one of these points. (Briefly explain.)

(d) (5 points) In the randomized incremental algorithm for computing a trapezoidal map, when the ith segment is added, *up to constant factors*, the expected number of newly created trapezoids is (select one):

- 1
- i
- $\log i$
- n/i
- $\log n$
- None of these. What is it?

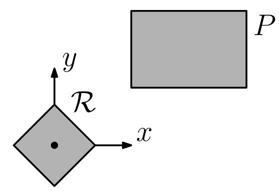


Figure 1: C-obstacle.

(e) (5 points) You are doing translational motion planning in $\mathbb{R}^2$. The obstacle P is an axis-aligned rectangle. The robot $\mathcal{R}$ is a 45° rotation of a square. What is the maximum number of edges possible in the C-obstacle $(P \oplus (-\mathcal{R}))$? (Briefly explain.)

Problem 2. (20 points) You are given a set $P = \{p_1, \ldots, p_n\}$ in $\mathbb{R}^2$. A *slab* $S(a, b^-, b^+)$ is the region between two pair of parallel lines, $S(a, b^-, b^+) = \{p : ap_x + b^- \le p_y \le ap_x + b^+\}$. The *height* of the slab is the vertical distance between the lines, $b^+ - b^-$.

Given a parameter k, where $0 \le k \le \frac{n-3}{2}$, we say the slab is k-*centered* if there are k points lying strictly above the slab and k points lying strictly below the slab (see Fig. 2). We are interested in the k-centered slab of minimum height.

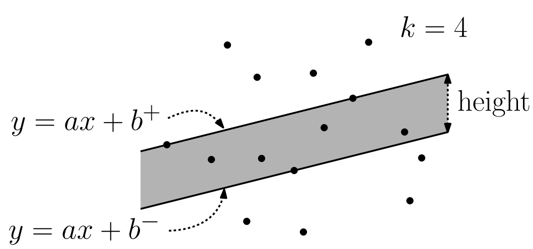


Figure 2: k-Centered slabs.

(a) (5 points) Assuming the standard dual transformation given in class $(a, b) \leftrightarrow y = ax - b$, describe the dual-equivalent formulation of this problem. That is, what is the slab? what is its height? what condition is satisfied for the slab be k-centered?

(b) (5 points) Assuming that P is in general position, prove that the minimum height k-centered slab has three points on its boundary, with two points on one line and one on the other.

(c) (10 points) Present an efficient algorithm, which given P and k ($0 \le k \le \frac{n-3}{2}$), computes the minimum-height k-centered slab. Derive your algorithm's running time and justify its correctness. (**Hint:** Plane sweep in the dual arrangement.)

Problem 3. (20 points) Explain how to use/modify range trees to answer the following queries. The input is an n-element point set P in $\mathbb{R}^2$.

(a) (10 points) A *right-triangle query* involves a region defined by a right triangle whose two legs are parallel to the coordinate axes and of *equal length*, such that the right angle is in the lower-left corner of the triangle. The query is defined by its lower left vertex $v = (v_x, v_y)$ and the length w of its two legs (see Fig. 3(a)). The answer is the number of points that lie within the triangle. Present the data structure and query algorithm. Derive its space usage and the query time.

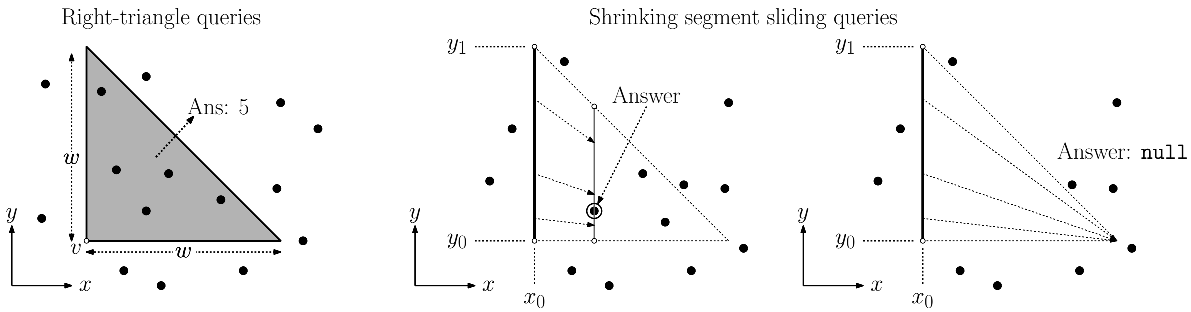


Figure 3: Range tree queries.

(b) (10 points) In a *shrinking segment-sliding query*, you are given a vertical line segment with x-coordinate x_0 and endpoints at y-coordinates y_0 and y_1, where $y_0 < y_1$. As this segment slides to the right it shrinks in height. The lower endpoint of the segment stays at $y = y_0$, but for each w units the segment slides horizontally, its height decreases by w. The answer to the query is the first point that is hit by the sliding-shrinking segment. If the segment slides so far that it shrinks to height zero, the answer is `null` (see Fig. 3(b)). Present the data structure and query algorithm. Derive its space usage and the query time.

Problem 4. (25 points) Consider the range space $\Sigma = (\mathbb{R}^2, \mathcal{T})$, where $\mathcal{T}$ is the set of all right triangles whose two legs are parallel to the coordinate axes, so that the right angle is in the lower-left corner of the triangle (see Fig. 4).

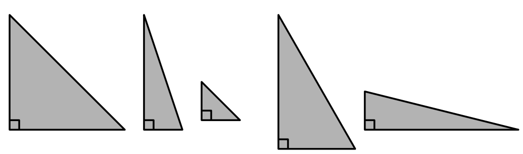


Figure 4: Set system of axis-aligned right triangles.

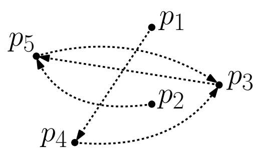
(a) (5 points) Give an example of a 4-element point set P in $\mathbb{R}^2$ that is shattered by Σ, and demonstrate why it is shattered. (For preciseness, indicate the coordinates of the points, but you can present a drawing to illustrate how to shatter them.)

(b) (12 points) Prove that no 5-element point set in $\mathbb{R}^2$ is shattered by Σ. (You can continue your answer on the top of the next page)

(c) (3 points) What is the VC-dimension of Σ? (No justification needed.)

(d) (5 points) Given a point set P in $\mathbb{R}^2$ with n points, give a (tight) asymptotic upper bound on the number of distinct subsets of P determined by Σ. (Using the notation given in class, this is $|\mathcal{T}_{|P}|$. No justification needed.)

Problem 5. (20 points) Given a set $P = \{p_1, \ldots, p_n\}$ of points in $\mathbb{R}^d$, for each $p_i \in P$ its *farthest neighbor* is the point $p_j \in P$ from P that is farthest from p_i.



i:	Index of p_i's farthest neighbor:
1	4
2	5
3	5
4	3
5	3

Figure 5: All farthest neighbors.

(a) (10 points) Prove that there must exist at least one pair of points $p_i, p_j \in P$ such that p_j is the farthest neighbor of p_i and vice versa. (For example, (p_3, p_5) above satisfies this.) Assume by general position that all inter-point distances are distinct.

(b) (10 points) Assuming that the points are in the plane, prove that the farthest neighbor of any point is a vertex of P's convex hull.

Problem 6: (15 points) Let $P = \{p_1, \ldots, p_n\}$ be a set of points in $\mathbb{R}^d$, and let Δ denote the diameter of P, that is, the distance between its farthest pair. We say that a pair $p_i, p_j \in P$ is *diametrical* if $\|p_j - p_i\| = \Delta$. (Let us ignore general position and imagine that there may be many diametrical pairs.)

Present an approximation algorithm that, given P, $\Delta = \mathrm{diam}(P)$, and $\varepsilon > 0$, counts the number of pairs that are approximately diametrical. This means that for any pair $p_i, p_j \in P$:

- if $\|p_j - p_i\| = \Delta$ then the pair *must* be counted, and
- if $\|p_j - p_i\| < \Delta/(1 + \varepsilon)$ then the pair *must not* be counted.

Otherwise, your algorithm is free to count or ignore the pair. Justify your algorithm's correctness and derive its running time. **Hint:** WSPDs. Your algorithm should run in time $O(n \log n + n/\varepsilon^d)$.