Sample Problems for the Final Exam

The final exam will be **Thu, Dec 16, 8:00am-10:00am** in class. The exam will be closed-book and closed-notes. You may use *two* sheets of notes (front and back). The following problems have been collected from old homeworks and exams. They do not necessarily reflect the actual difficulty or coverage of questions on the final exam. The final will be comprehensive, but will emphasize material since the midterm.

In all problems, unless otherwise stated, you may assume general position, and you may use of any results presented in class or any well-known result from algorithms and data structures.

Problem 1. Give a short answer (a few sentences) to each question. Unless explicitly requested, explanations are not required, but may be given for partial credit.

(a) A *dodecahedron* is a convex polyhedron that has 12 faces, each of which is a 5-sided pentagon. Every vertex has degree 3. How many vertices and edges does the dodecahedron have? Show how you derived your answer.

(b) Given a set P of n points in the plane, what is the maximum number of edges in P's Voronoi diagram? (For full credit, express your answer up to an additive constant.)

(c) When the ith site is added to the Delaunay triangulation using the randomized incremental algorithm, what is the worst-case number of edges that can be incident on the newly added site? What can you say about the expected-case number of such edges (assuming that points are inserted in random order)?

(d) For each of the following assertions about the Delaunay triangulation of a set P of n points in the plane, which are True and which are False?

(i) The Delaunay triangulation is a t-spanner, for some constant t

(ii) The Euclidean minimum spanning tree of P is a subgraph of the Delaunay triangulation

(iii) Among all triangulations of P, the Delaunay triangulation maximizes the minimum angle

(iv) Among all triangulations of P, the Delaunay triangulation minimizes the maximum angle

(v) Among all triangulations of P, the Delaunay triangulation minimizes the total sum of edge lengths

(e) An arrangement of n lines in the plane has exactly n^2 edges. How many edges are there in an arrangement of n planes in 3-dimensional space? (Give an exact answer for full credit or an asymptotically tight answer for half credit.) Explain briefly.

(f) Let P and Q be two simple polygons in $\mathbb{R}^2$, where P has m vertices and Q has n vertices. What is the maximum number of vertices on the boundary of the Minkowski sum $P \oplus Q$ (asymptotically) assuming:

(i) P and Q are both convex

(ii) P is convex but Q is arbitrary

(iii) P and Q are both arbitrary

(g) In each of the following cases, what is the asymptotic worst-case complexity (number of vertices) on the boundary of the union of n of the following objects in $\mathbb{R}^2$:

(i) axis-parallel squares

(ii) axis-parallel rectangles (of arbitrary heights and widths)

(iii) rectangles (of arbitrary heights and widths which need not be axis parallel)

(iv) axis-parallel rectangles, where the width to height ratio is either 4×1 or 1×4.

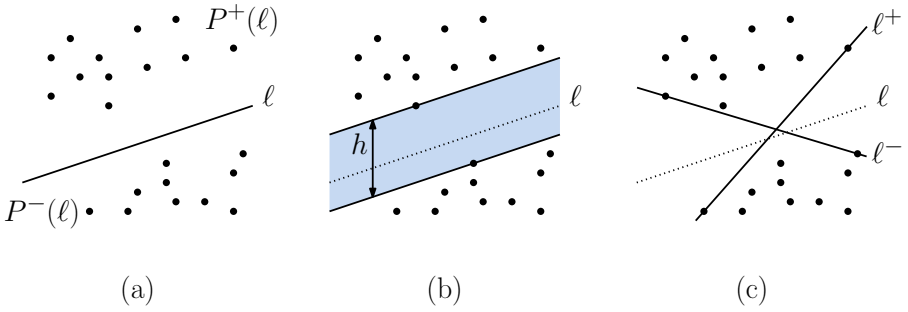

Figure 1: Query problem.

Problem 2. You are given a set P of n points in $\mathbb{R}^2$. A nonvertical line ℓ partitions P into two (possibly empty) subsets: $P^+(\ell)$ consists of the points lie on or above ℓ and $P^-(\ell)$ consists of the points of P that lie strictly below ℓ (see Fig. 1(a)).

Given the point set P, present data structures for answering the following two queries. In each case, the data structure should use $O(n^2)$ space, it should answer queries in $O(\log n)$ time. (You do not need to explain how to build the data structure, but it should be constructable in polynomial time in n.)

(a) The input to the query is a nonvertical line ℓ. The answer is the maximum vertical distance h between two lines parallel to h that lie between $P^+(\ell)$ and $P^-(\ell)$ (see Fig. 1(b)). For simplicity, you may assume that neither set is empty (implying that h is finite).

(b) Again, the input to the query is a nonvertical line ℓ. The answer to the query are the two lines ℓ^- and ℓ^+ of minimum and maximum slope, respectively, that separate $P^+(\ell)$ from $P^-(\ell)$ (see Fig. 1(c)). You may assume that $P^+(\ell)$ from $P^-(\ell)$ are *not* separable by a vertical line (implying that these two slopes are finite).

Problem 3. You are given a set P of n points in the plane and a path π that visits each point exactly once. (This path may self-intersect. See Fig. 2.)

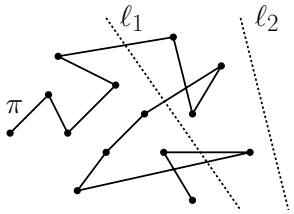

Figure 2: Path crossing queries.

Explain how to build a data structure from P and π of space $O(n)$ so that given any query line ℓ, it is possible to determine in $O(\log n)$ time whether ℓ intersects the path. (For example, in Fig. 2 the answer for ℓ_1 is "yes," and the answer for ℓ_2 is "no.") (Hint: Duality is involved, but the solution requires a bit of lateral thinking.)

Problem 4. Consider the following two geometric graphs defined on a set P of points in the plane.

(a) *Box Graph:* Given two points $p, q \in P$, define $\text{box}(p, q)$ to be the square centered at the midpoint of $\overline{pq}$ having two sides parallel to the segment $\overline{pq}$ (see Fig. 3(a)). The edge (p, q) is in the box graph if and only if $\text{box}(p, q)$ contains no other point of P (see Fig. 3(b)). Show that the box graph is a subgraph of the Delaunay triangulation of P.

(b) *Diamond Graph:* Given two points $p, q \in P$, define diamond(p, q) to be the square having $\overline{pq}$ as a diagonal (see Fig. 3(c)). The edge (p, q) is in the diamond graph if and only if diamond(p, q) contains no other point of P (see Fig. 3(d)). Show that the diamond graph may not be a subgraph of the Delaunay triangulation of P. (Hint: Give an example that shows that the diamond graph is not even planar.)

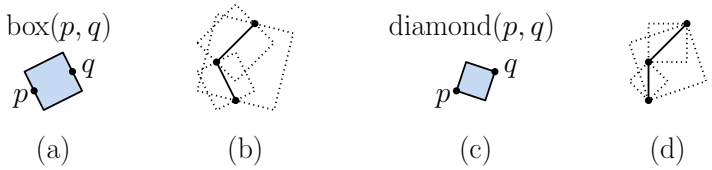

Figure 3: The box and diamond graphs.

Problem 5. Consider the range space $(P, \mathcal{R})$ where P is a set of n points in the plane, and $\mathcal{R}$ is the set of all ranges arising by intersecting P with a closed halfplane.

(a) Show that the VC-dimension of halfplane ranges is at least three by giving an example of a set of three points in the plane that are shattered by the set of halfplane ranges.

(b) Show that the VC-dimension of halfplane ranges is at most three, by proving that no four-element set can be shattered by halfplane ranges.

(c) Prove from first principles that $|\mathcal{R}| = O(n^2)$, where $n = |P|$. You are *not* allowed to appeal to Sauer's lemma. (Hint: Explain how to map each range to one of $O(n^2)$ *canonical halfplanes*, containing the same set of points as the original halfplane.)

Problem 6. (Here is another problem on VC-dimension from another semester.)

In this problem we will consider the VC-dimension of two simple range spaces. Define a *quad* to be a four-sided polygon that is bounded to the left and right by vertical sides, on the bottom by a horizontal side, and the slope of the top side is arbitrary (see Fig. 4(a)). Define a *restricted quad* to be a quad whose left side is at $x = 0$, whose right side is at $x = 1$, and whose bottom side is at $y = 0$ (see Fig. 4(b)).

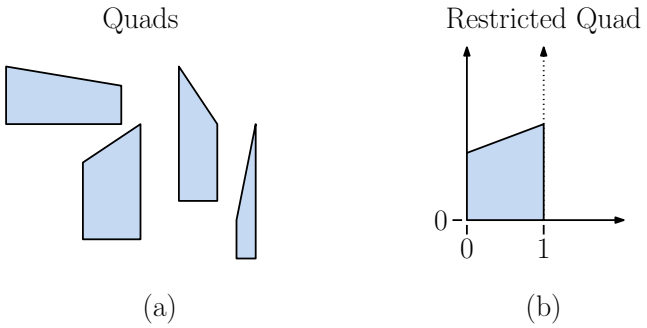

Figure 4: Quads and restricted quads.

(a) Prove that the VC-dimension of *restricted quads* is at least 2 by showing that there exists a 2-element point set in $\mathbb{R}^2$ that is shattered by the set of restricted quads.

(b) Prove that the VC-dimension of *restricted quads* is at most 2 by showing that no point 3-element point set in $\mathbb{R}^2$ is shattered by the set of restricted quads. (Hint: Label the three points p_1, p_2,

and p_3 from left to right. There are two cases depending on whether p_2 lies above or below the segment $\overline{p_1 p_3}$.)

(c) Prove that the VC-dimension of (general) *quads* is at least 5 by showing that there exists a 5-element point set in $\mathbb{R}^2$ that is shattered by the set of quads.

(d) Prove that the VC-dimension of (general) *quads* is at most 5 by showing that no point 6-element point set in $\mathbb{R}^2$ is shattered by the set of restricted quads. (Hint: A careful proof with full details will take too long. It suffices to briefly explain how to generalize your answer to part (b).)

Problem 7. Given a set of n points P in $\mathbb{R}^d$, and given any point $p \in P$, its *nearest neighbor* is the closest point to p among the remaining points of P. Note that nearest neighbors are not reflexive, in the sense that if p is the nearest neighbor of q, then q is not necessarily the nearest neighbor of p. Given an approximation factor $\varepsilon > 0$, we say that a point $p' \in P$ is an *ε-approximate nearest neighbor* to p if $\|pp'\| \leq (1+\varepsilon)\|pp''\|$, where p'' is the true nearest neighbor to p.

Show that in $O(n \log n + (1/\varepsilon)^d n)$ time it is possible to compute an ε-approximate nearest neighbor for every point of P. Justify the correctness of your algorithm. Hint: This can be solved using either WSPDs or spanners.

Note: There exists an algorithm that runs in $O(n \log n)$ time that solves this problem exactly, but it is considerably more complicated than the one I have in mind here.

Problem 8. A set P of n points in $\mathbb{R}^d$ determines a set of $\binom{n}{2}$ different distances. Define $\Delta(P)$ to be this set of distances $\{\|p_i - p_j\| : 1 \leq i < j \leq n\}$. Given an integer k, where $1 \leq k \leq \binom{n}{2}$, we are interested in computing the kth smallest distance from this set. Normally, this would take $O(n^2)$ time, so let's consider a fast approximation algorithm.

Let $\delta(P,k)$ denote the exact kth smallest distance in $\Delta(P)$. Given $\varepsilon > 0$, a distance value x is an ε-approximation to $\delta(P,k)$ if

$$\frac{\delta(P,k)}{1+\varepsilon} \;\leq\; x \;\leq\; (1+\varepsilon)\delta(P,k).$$

Present an efficient algorithm to compute such a value x. Justify your algorithm's correctness and derive its running time. (**Hint:** Use well-separated pair decompositions. You may assume that, when the quadtree is computed, each node u of the quadtree is associated with an integer $\mathrm{wt}(u)$, which indicates the number of points of P lying within u's subtree.)

You may assume that d and ε are constants (independent of n). I know of an algorithm that runs in time $O(n \log n + n/\varepsilon^d)$ time, but I will accept for full credit an algorithm that runs in time $O((n \log n)/\varepsilon^d)$.

Problem 9. You are given a set P of n points in $\mathbb{R}^d$ and an approximation factor $\varepsilon > 0$. An (exact) *distance query* is defined as follows. You are given a real $\delta > 0$, and you are to return a count of all the pairs of points $(p,q) \in P \times P$, such that $\|pq\| \geq \delta$. In an *ε-approximate distance query*, your count *must* include all pairs (p,q) where $\|pq\| \geq \delta(1+\varepsilon)$ and it *must not* include any pairs (p,q) where $\|pq\| < \delta/(1+\varepsilon)$. Pairs of points whose distances lie between these two bounds may or may not be counted, at the discretion of the algorithm.

Explain how to preprocess P into a data structure so that ε-approximate distance counting queries can be answered in $O(n/\varepsilon^d)$ time and $O(n/\varepsilon^d)$ space. (Hint: Use a well-separated pair decomposition. Explain clearly what separation factor is used and any needed modification to the WSPD construction.)

Problem 10. This problem considers motion planning in a dynamic setting, which is inspired by various old video games. You are given a *robot* that consists of a line segment of unit length that resides on the x-axis. The robot can move left or right (but not up or down) at a speed of up to one unit per second. You are given two real values x^- and x^+, and the robot must remain entirely between these two values at all times (see Fig. 5). The robot's *reference point* is its left endpoint, and at time $t = 0$, the left endpoint is located at x^-. (You may assume that $x^+ > x^- + 1$.)

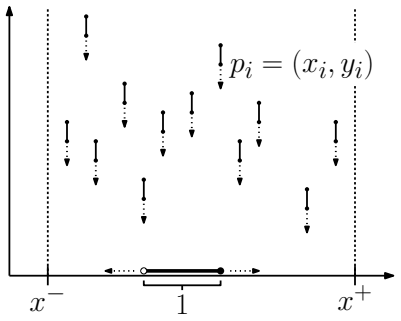

Figure 5: Robot motion planning.

You are also given a set of *missiles* in the form of n vertical line segments, each of length 0.2, that fall down from the sky at a rate of two units per second. Each of these vertical segments is specified by the coordinates of its lower endpoint at time $t = 0$. So, if $p_i = (x_i, y_i)$ is the starting position of the ith missile, then at time t is its lower endpoint is located at $(x_i, y_i - 2t)$, and its upper endpoint is at $(x_i, y_i - 2t + 0.2)$. You may assume that $x^- \leq x_i \leq x^+$.

The question is whether it is possible for the robot to move in a manner to avoid all the missiles. We will explore an algorithm for solving this problem.

(a) A natural way to define the robot's configuration at any time is as a pair (t, x), where t is the current time, and x is the location of the robot's left endpoint. Based on this, what is the C-obstacle associated with a missile whose starting position is p_i (as defined above)? In other words, describe the set of robot configurations (t, x) such that the robot intersects this missile. (Please provide low-level details, as opposed, say, to expressing this as a Minkowski sum.)

(b) Provide a complete characterization of the properties of a path in configuration space (assuming it exists) that corresponds to a motion plan for the robot that satisfies the robot's speed constraints and avoids all the missiles. Be sure to include constraints on the path's starting and ending positions and include the robot's maximum speed.