CMSC 754: Final Exam

In all problems, unless otherwise stated, you may assume that inputs are in *general position.* If you are asked for an $O(T(n))$ time algorithm, you may give a *randomized* algorithm with *expected* time $O(T(n))$. If you are asked to give an algorithm, also explain how it works and derive its running time. (Unless otherwise stated, formal proofs of correctness are not required.)

Problem 1. (30 points) Give a short answer (a few sentences at most) to each question. Except where requested, explanations are not required.

(a) (5 points) Given a simple polygon with n sides, what can be said about the number of triangles in any triangulation? Either state the exact number as a function of n, or provide upper and lower bounds.

(b) (8 points) In class we showed that the *trapezoidal map* of n nonintersecting line segments has at most $6n + 4$ vertices and at most $3n + 1$ trapezoids. Suppose instead that there are k instances where two segments intersect. (In Fig. 1 there are $n = 3$ segments and $k = 2$ intersection points.) At each intersection point, we shoot two bullet paths, up and down. As a function of n and k, what is the (exact) number of vertices and trapezoids in the resulting trapezoidal map? (No explanation needed.)

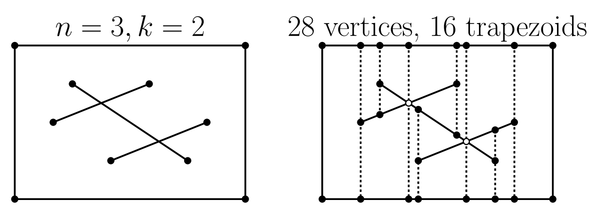

Figure 1: A trapezoidal map with $n = 3$ segments and $k = 2$ intersection points.

(c) (4 points) In our backward analysis of randomized incremental algorithms, we were careful to argue that the structure being analyzed does *not* depend on the order of insertion. What aspect of the analysis would *fail* if the structure did depend on the order of insertion? (Explain briefly.)

(d) (5 points) Consider the portion of the line arrangement shown in Fig. 2. Illustrate the *zone* of the line ℓ in this arrangement.

(e) (8 points) Suppose you have a workspace with n disjoint obstacles, each of which is a k-sided convex polygon, and you are given a robot that translates in this workspace, which is modeled as an m-sided convex polygon.

   (i) As a function of n, k, and m, give an upper bound on the total number of vertices in all the C-obstacles? (You may express your answer using big-Oh notation.) Explain briefly.

   (ii) The C-obstacles may overlap. As a function of n, k, and m, give an upper bound on the total number of vertices in the *union* of the C-obstacles? (Again, you can use big-Oh notation.) Explain briefly.

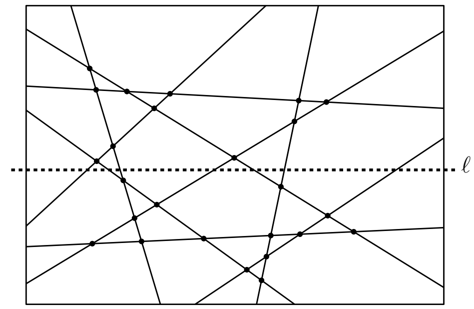

Figure 2: The zone of an arrangement.

Problem 2. (20 points) Present LP solutions to the following problems. In each case, explain how the problem is formulated as an instance of LP (and what the dimension of the space is), and how the result of the LP (feasible, infeasible, unbounded) is to be interpreted in answering the problem.

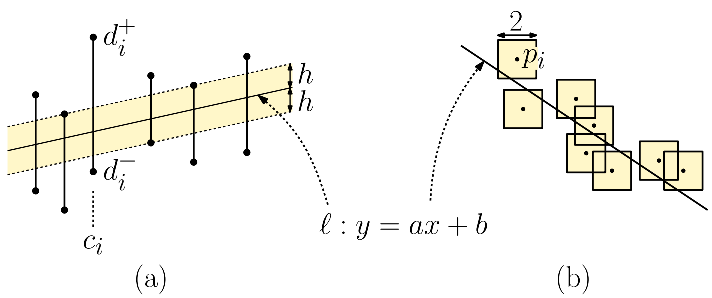

Figure 3: Stabbing segments and squares.

(a) (10 points) You are given a set of n vertical line segments in the plane $S = \{s_1, \ldots, s_n\}$, where each segment s_i is described by three values, its x-coordinate c_i, its upper y-coordinate d_i^+ and its lower y-coordinate d_i^-. Compute a line $\ell : y = ax + b$ that intersects all of these segments. If it exists, return the line that maximizes the vertical spacing h above and below the line (see Fig. 3(a)).

(b) (10 points) You are given a collection of n axis-aligned squares in the plane, each of side length 2. The squares are centered at the points $P = \{p_1, \ldots, p_n\}$, where $p_i = (c_i, d_i)$ (see Fig. 3(b)). Determine whether there exists a line $\ell : y = ax + b$ that intersects all of these squares. If it exists, return any such line. **Hint:** You may need to make multiple calls to LP.

Problem 3. (30 points) For each of the following search problems, explain how to map this to a data structure that we have seen this semester. In each case, explain which data structure you will use (e.g., point-location, kd-tree, range tree, etc.), what information is stored in the data structure, how much space it uses, and what the query time is. There may be multiple options, and if so, select one that is most efficient.

The goal is to achieve space that is either linear in n or slightly higher (e.g., $O(n)$, $O(n \log n)$, or $O(n \log^2 n)$), and to achieve a query time that $O(\log n)$ or slightly higher (e.g., $O(\log^2 n)$, or $O(\log^3 n)$). We will give partial credit for a correct answer, even if the space or query time is not optimal.

Hint: Keep your answers brief. We are looking for a reduction to known data structure, perhaps with additional minor modifications.

(a) (10 points) The data is a set of n points P in $\mathbb{R}^2$. The query is an axis-aligned rectangle $Q = [x_0, x_1] \times [y_0, y_1]$. The answer to the query is the area of the smallest axis-aligned rectangle that contains all the points of $P \cap Q$ (see Fig. 4(a)).

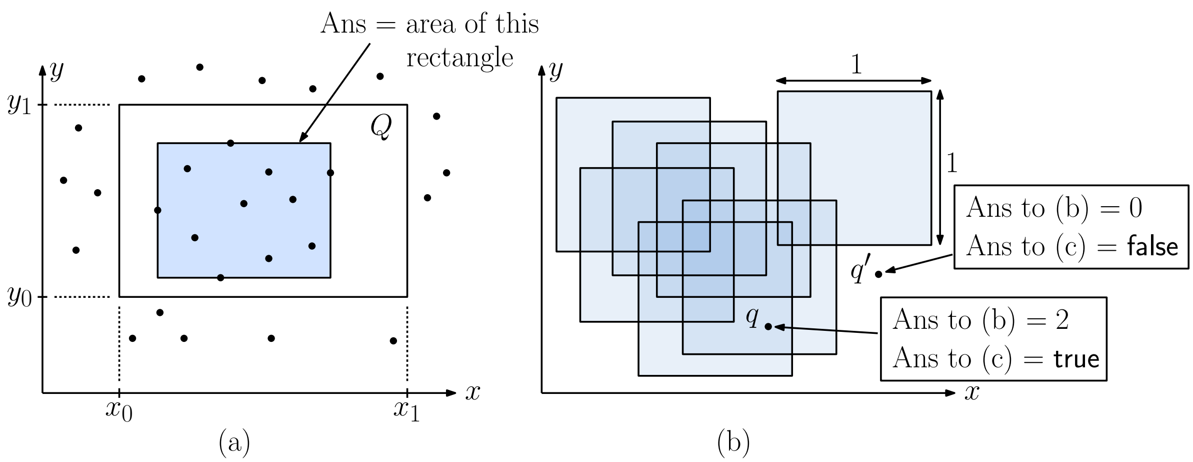

Figure 4: Geometric queries.

(b) (10 points) The data is a set S of n axis-aligned unit squares in $\mathbb{R}^2$ (that is, each has side length 1). The query is a point $q = (q_x, q_y)$. The answer to the query is the number of squares of S that contain q (see Fig. 4(b)).

(c) (10 points) The data set is again a set S of n axis-aligned unit squares in $\mathbb{R}^2$ (that is, each has side length 1). The query is a point $q = (q_x, q_y)$. The answer to the query is true if q lies in the union of the squares and false otherwise (see Fig. 4(b)).

Problem 4. (20 points) For each of the following range spaces, derive its VC-dimension and prove your result. (Note that in order to show that the VC-dimension is k, you need to give an example of a k-element subset that is shattered and prove that no set of size $k + 1$ can be shattered.) Throughout, you may assume that points are in general position.

Hint: Proving the upper bound on the VC-dimension can involve multiple cases. Do your best to explain what the relevant cases are, but don't bother proving each one formally. For each case, you can give a short one-sentence explanation (or draw a picture) to explain which subsets cannot be generated.

(a) (8 points) $\Sigma = (\mathbb{R}^2, \mathcal{S})$, where $\mathcal{S}$ consists of parallelograms having two horizontal sides and two sides that have a slope of 1 (see Fig. 5(a)).

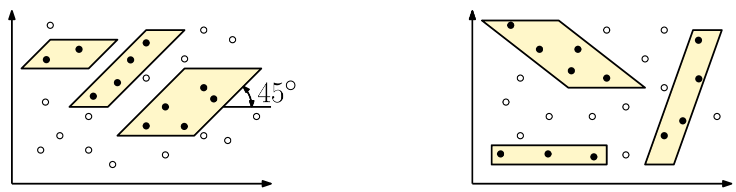

Figure 5: VC-Dimension of some range spaces.

(b) (12 points) $\Sigma = (\mathbb{R}^2, \mathcal{P})$, where $\mathcal{P}$ consists of parallelograms having two horizontal sides and two sides of arbitrary (nonzero) slope (see Fig. 5(b)).

Problem 5. (20 points) You are given a set P of n points in $\mathbb{R}^d$ and a real $\delta > 0$. The objective of a δ-*distance query* is to return a count of all the pairs of distinct points $x, y \in P$, such that $\|x - y\| \leq \delta$. Given $0 < \varepsilon < 1$, in an ε-*approximate* δ-*distance query*, your count *must* include all pairs such that $\|x - y\| \leq \delta$, and it *must not* include any pair such that $\|x - y\| > (1 + \varepsilon)\delta$. 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 what modifications are needed to the WSPD construction. It is not necessary to derive the tightest possible bound on the separation factor, but formally justify your choice.