CMSC 754: Midterm Exam

This exam is closed-book and closed-notes. You may use one sheet 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 100 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. (20 points; 4–6 points each) Short-answer questions.

(a) In our plane-sweep algorithm for computing line segment intersections, we were careful to store only those intersection points involving pairs of segments that are *adjacent on the current sweep line.* Why did we do this?

(b) You are given four points a, b, c, d in $\mathbb{R}^2$. Using just orientation tests, show how to test whether the line segment $\overline{ab}$ intersects the line segment $\overline{cd}$. (Briefly explain.)

(c) True or False: If a simple polygon is both x-monotone and y-monotone, then it is monotone with respect to any direction. (Briefly explain your answer.)

(d) Consider the line arrangement shown in the figure below. Suppose that we insert the line ℓ_i into this arrangement. Indicate (by redrawing the figure) which edges of the arrangement are traversed by the insertion algorithm presented in class.

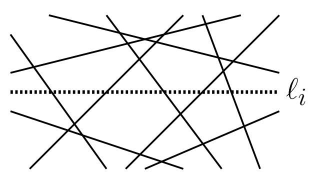

Figure 1: Inserting a line into an arrangement.

Problem 2. (15 points) The objective of this problem is to explore independent sets in triangulations. Throughout this problem, let $P = \{p_1, \ldots, p_n\}$ denote a set of n sites in the plane, and let $T(P)$ denote an arbitrary (not necessarily Delaunay) triangulation of P (see Fig. 2(a)). Define the degree of any site $p \in P$, denoted $\deg(p)$, to be the number of edges of $T(P)$ incident on it.

(a) (5 points) Prove that there exists a constant c, such that $\sum_{p \in P} \deg(p) \leq cn$, for all sufficiently large n.

(Recall that if there are h points on the convex hull, there are $2n - h - 2$ triangles and $3n - h - 3$ edges. Ideally, your answer should apply for any value of h, but for partial credit, you may assume that h has a specific value of your choosing.)

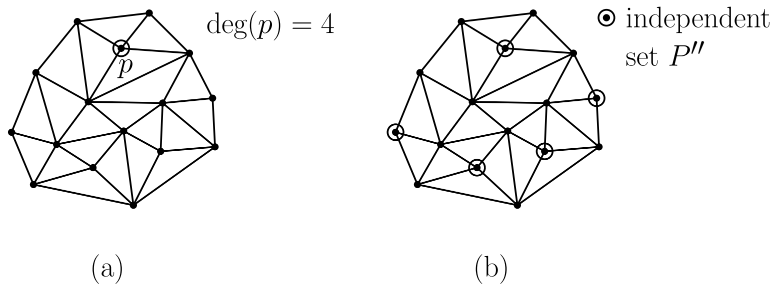

Figure 2: Independent sets in a triangulation.

(b) (5 points) Let c be the constant derived in your solution to (a). We say that a site $p \in P$ has *high degree* if $\deg(p) \geq 3c$, and otherwise it has *low degree*. Let $P' \subseteq P$ be the subset of low degree sites of P. Prove that there exists a constant c' (which may depend on c) such that, for all sufficiently large n, $|P'| \geq n/c'$.

(c) (5 points) Define an *independent set* to be a subset $P'' \subseteq P$ such that no two sites in P'' are adjacent in $T(P)$ (see Fig. 2(b)). Given the previous constants c and c', prove that there exists a constant $c'' > 1$ (depending on c and c') such that, for all sufficiently large n, P contains an independent set of size at least n/c'' *consisting entirely of low-degree sites.*

Problem 3. (30 points) In parts (a) and (c) below, you are asked to give a reduction to linear programming (LP). 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.

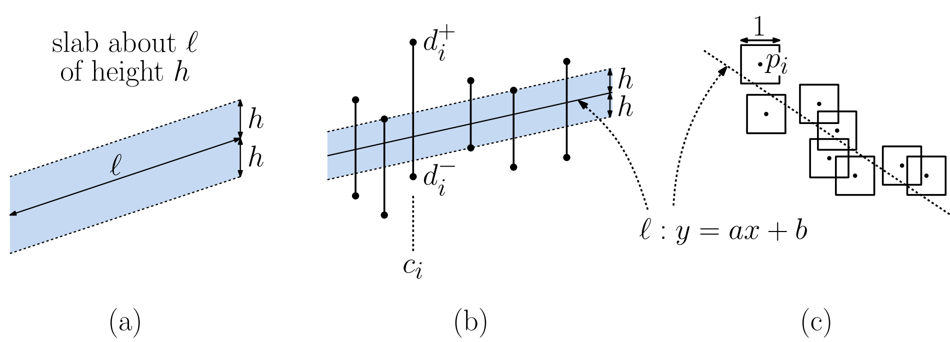

Figure 3: Stabbing segments.

(a) (10 points) Given a line ℓ, define the *slab* of height h centered about ℓ to be the region bounded between the two lines parallel to ℓ, one h units above and on h units below (see Fig. 3(a)).

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^- (see Fig. 3(b)).

Apply LP to determine whether there exists a line $\ell : y = ax + b$ that intersects all of these segments. Further, if such a line exists, return the line ℓ with the property that it is the center of the slab of *maximum height* h that cuts through all the segment. (You can solve just the existence problem for partial credit.)

(b) (10 points) Suppose that your LP from part (a) reveals that there is no line that stabs all the segments. Instead, you decide to solve the following optimization problem. Given a set of vertical line segments S (as in part (a)), find a line $\ell : y = ax - b$ that intersects the *maximum* number of segments of S.

You decide to solve this problem in the dual setting. Using the dual transformation given in class, explain what the *equivalent optimization problem* is in the dual setting. (That is, explain how to dualize the line segments of S, how to dualize the line ℓ, and what property the dual point ℓ^* must satisfy so that we are effectively solving the same optimization problem.) You do *not* need to explain how to solve this dual problem.

(c) (10 points) You are given a collection of n axis-aligned unit squares in the plane. The squares are centered at the points $P = \{p_1, \ldots, p_n\}$, where $p_i = (c_i, d_i)$ (see Fig. 3(c)).

Apply LP to determine whether there exists a line $\ell : y = ax + b$ that intersects all of these squares. If it exists, return any such line.

Problem 4. (20 points) In this problem, we will consider two query problems involving a set of n circular disks in the plane (which may overlap), each of unit radius. Let $P = \{p_1, \ldots, p_n\}$ denote their centers, and let us assume that at least one of these disks contains the origin O.

For each of the parts below, explain how to preprocess these disks into a data structure to answer the specified query. In each case, your data structure should use $O(n)$ space, be constructed in $O(n \log n)$ time, and answer queries in $O(\log n)$ time. Briefly justify the correctness and running times of your solutions.

(a) (15 points) Given a query point q, determine whether it is possible to move q to the origin, so that the path lies *entirely within the union of these disks*. For example, in Fig. 4(a), q_1 can reach the origin O but q_2 cannot. Note that if q does not lie within any disk, the answer is trivially "no".

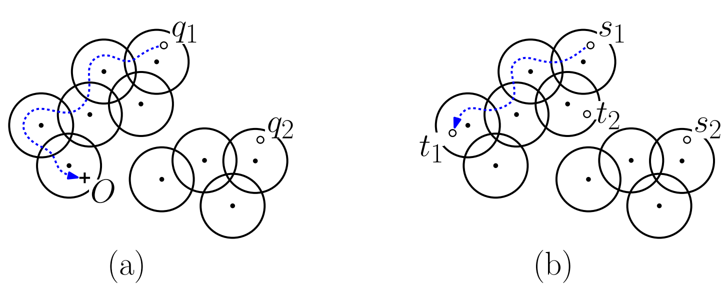

Figure 4: Motion planning among disks.

(b) (5 points) Given two query points s and t, determine whether it is possible to move from s to t, so that the path lies entirely within the union of these disks. For example, in Fig. 4(b), s_1 can reach t_1, but s_2 cannot reach t_2. (Hint: You can explain the changes you would make to the solution from (a).)

Problem 5. (15 points) This problem is inspired from applications in surveillance. Given a simple polygon P, we say that two points p and q are *visible* to each other if the open line segment between them lies entirely within P's interior. We allow for p and q to lie on P's boundary, but the segment between them cannot pass through any vertex of P (see Fig. 5(a)).

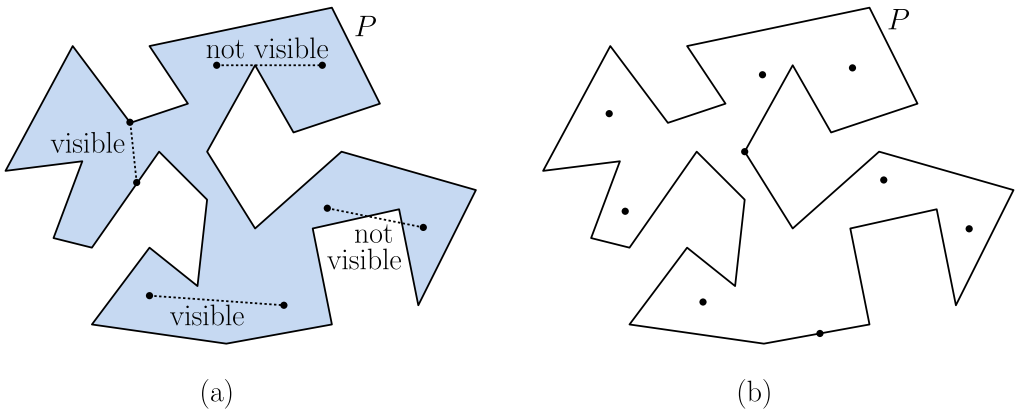

Figure 5: (a) Visibility and (b) a guarding set of size 9 for P.

A *guarding set* for P is any set of points G, called *guards*, lying in P (either on its boundary or in its interior) such that every point in P's interior is visible to at least one guard of G. Note that guards may be placed on vertices, along edges, or in P's interior (see Fig. 5(b)).

Prove that there exists a constant $c \geq 1$ such that (for all sufficiently large n) every n vertex simple polygon P has a guarding set of size at most n/c. For full credit, show that $c = 3$ works. For partial credit, show that some smaller value of c (e.g., $c = 2$) works. You do *not* need to show how to compute this set. (Hint: Decompose the polygon into simpler pieces.)