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 in the exam booklet. 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) Consider the two segments s_1 and s_2 shown in Fig. 1(a).

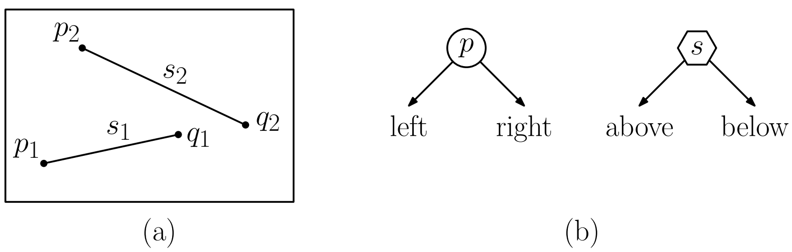


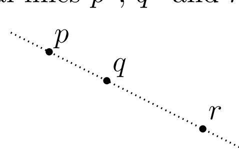
Figure 1: Trapezoidal map and point location.

(a) (5 points) Show the (final) *trapezoidal map* for these two segments, assuming the insertion order $\langle s_1, s_2 \rangle$. Draw over the figure below.

(b) (15 points) Show the *point-location data structure* resulting from the construction given in class, assuming the insertion order $\langle s_1, s_2 \rangle$. Recall the nodes types in Fig. 1(b). (Note: We will give 50% partial credit if your data structure works correctly, even if it is not the same as the one from class.)

Problem 2. (20 points; 4–6 points each) Short-answer questions.

(a) You have three distinct, collinear points in the plane that appear in the left-to-right order p, q, r (see the figure below). Assume the dual transformation given in class, which maps the point (a, b) to the line $y = ax - b$. What can you assert about the relationship between the dual lines p^*, q^* and r^*? (Be as specific as possible.)



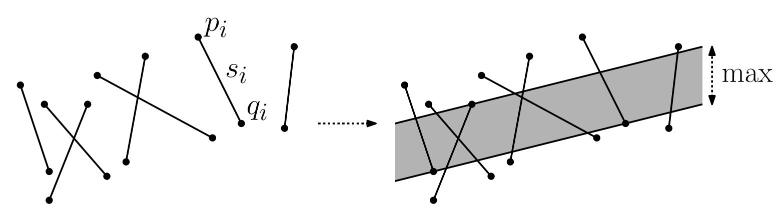
(b) You have two convex polygons P and Q, each having exactly n vertices. No two edges of P and Q are collinear. As a function of n, what is the maximum number of times the boundaries of P and Q can intersect? (No proof needed.) For $n = 5$, give an example (two convex pentagons) that illustrates your bound.

(c) Consider the linear-programming algorithm given in class for n constraints in dimension 2. In class we showed that the *expected-case* running time of the algorithm is $O(n)$. What is the *worst-case* running time of the algorithm? Briefly justify your answer (in a sentence or two).

(d) It is a fact that if P is a uniformly distributed random set of n points in a circular disk in the plane, the expected number of vertices of P's convex hull is $\Theta(n^{1/3})$. That is, the lower and upper bounds are both within some constant of $n^{1/3}$ for large n.

What is the average-case running time of Jarvis's algorithm for such an input? (If you forgot the running time of Jarvis's algorithm, we will give it to you for a 50% penalty on this problem.)

Problem 3. (15 points) You are given a set of line segments $S = \{s_1, \ldots, s_n\}$ in the plane, where $s_i = \overline{p_i, q_i}$ (see the figure below). A *slab* is the region of the plane between two parallel lines. The *vertical width* of the slab is the length of the slab's intersection with the y-axis. Present an efficient algorithm to compute the slab of maximum vertical width that stabs all the segments, so that for all i, p_i lies above the slab and q_i lies below the slab. If no such slab exists, your algorithm should indicate this. (**Hint:** This is possible in $O(n)$ time, but $O(n \log n)$ is acceptable for partial credit.)



Problem 4. (20 points) You are given an axis-parallel rectangle R in the plane and a set of line segments $S = \{s_1, \ldots, s_n\}$ that lie within R. The segments may intersect only at their endpoints, and their endpoints may lie on the boundary of R (see Fig. 2(a)).

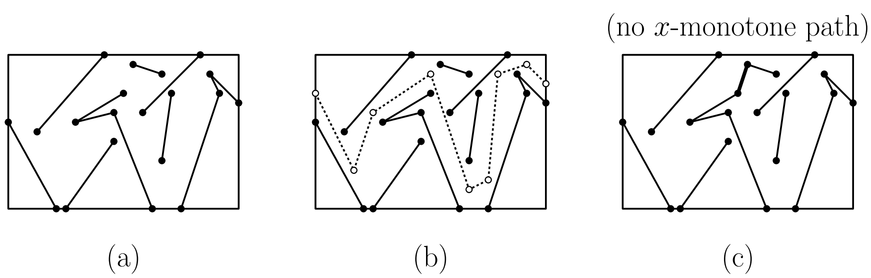


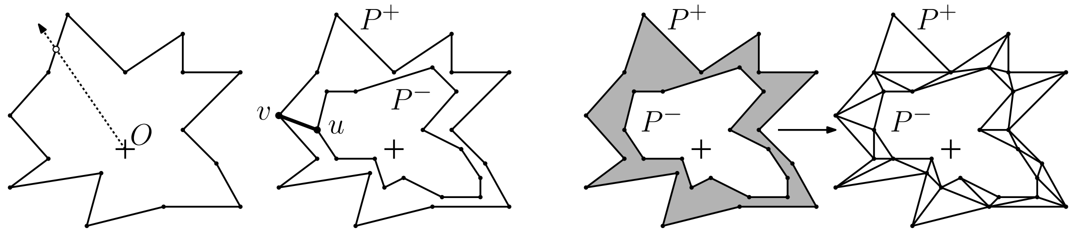
Figure 2: Monotone path.

Recall that a polygonal chain is *x-monotone* if any vertical line intersects the chain in at most one point. Present an algorithm which, given R and S, determines whether there exists an x-monotone polygonal chain, that does not intersect the interior of any segment, that starts anywhere on the left side of R, and goes to anywhere on the right side of R. Your algorithm *does not* need to output the path, simply "yes" or "no." (For example, the input from Fig. 2(b) the answer is "yes" as illustrated by the dotted path, and for the input from

Fig. 2(c) the output is "no".) Briefly justify your algorithm's correctness and derive its running time.

(**Hints:** For full credit, your algorithm should run in $O(n \log n)$ time. You may modify any algorithm from class.)

Problem 5. (25 points) A polygonal chain in the plane is *cyclically monotone* with respect to a point O if every ray emanating from O intersects the chain in a single point (see Fig. 3(a)). Henceforth, let us take O to be the origin of the coordinate system.



(a) (b) (c)

Figure 3: Triangulating and cyclically monotone region.

You are given two simple polygons, P^- and P^+, whose boundaries are cyclically monotone. These two chains do not intersect each other, and P^- is nested within P^+. Let n denote the total number of vertices on P^- and P^+.

(a) (10 points) Define a *cross diagonal* to be a line segment $\overline{uv}$, where u is a vertex of P^-, v is a vertex of P^+, and this segment lies entirely within the region between P^- and P^+ (see Fig. 3(b)). Present an efficient algorithm, which given just P^- and P^+, computes any one cross diagonal $\overline{uv}$. Briefly justify your algorithm's correctness and derive its running time. (**Hint:** This is possible in $O(n)$ time, but $O(n \log n)$ is acceptable for partial credit.)

(b) (15 points) Give an algorithm to triangulate the region between P^- and P^+ (see Fig. 3(c)). Briefly justify your algorithm's correctness and derive its running time. (**Hint:** This is possible in $O(n)$ time, but $O(n \log n)$ is acceptable for partial credit. It may help to assume that you are given a cross diagonal to start. You may modify an algorithm given in class.)