CMSC 754: Fall 2023 Dave Mount

Sample Problems for the Midterm Exam

The midterm exam will be this **Thursday, Oct 19 in class**. It will be *closed-book* and *closed-notes*, but you may use *one sheet of notes* (front and back).

Unless otherwise stated, you may assume *general position*. If you are asked to present an $O(f(n))$ time algorithm, you may present a *randomized algorithm* whose expected running time is $O(f(n))$. For each algorithm you give, derive its running time and justify its correctness.

Disclaimer: The following sample problems have been collected from old homeworks and exams. Because the material and order of coverage varies each semester, these problems *do not* necessarily reflect the actual length, coverage, or difficulty of the midterm exam.

Problem 0. Expect a problem asking you to work through all or part of an algorithm that was presented in class on a specific example.

Problem 1. Give a short answer to each question (a few sentences suffice).

(a) Explain how to use *at most three* orientation tests to determine whether a point d lies within the interior of a triangle $\triangle abc$ in the plane. You do *not* know whether $\triangle abc$ is oriented clockwise or counterclockwise (but you may assume that the three points are not collinear).

(b) Let P be a simple polygon with n sides, where n is a large number. As a function of n, what is the maximum number of *scan reflex vertices* that it might have? Draw an example to illustrate.

(c) A convex polygon P_1 is enclosed within another convex polygon P_2 (see Fig. 1(a)). Suppose you dualize the vertices of each of these polygons (using the dual transform given in class, where the point (a, b) is mapped to the dual line $y = ax - b$). What can be said (if anything) about the relationships between the resulting two sets of dual lines.

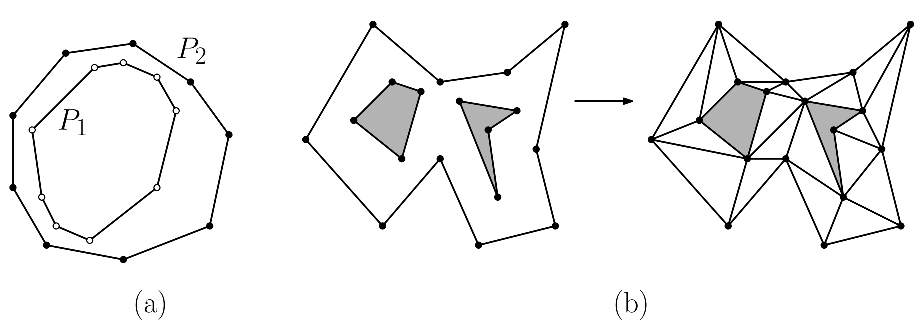


Figure 1: (a) Duals of nested polygons and (b) triangulating a polygon with holes.

(d) Any triangulation of any n-sided simple polygon has exactly $n - 2$ triangles. Suppose that the polygon has h polygonal holes each having k sides. (In Fig. 1(b), $n = 10$, $h = 2$, and $k = 4$). As a function of n, h and k, how many triangles will such a triangulation have? Explain briefly.

(e) You are given a set of n disjoint line segments in the plane that have m intersection points. (For example, in Fig. 2, $n = 4$ and $m = 3$). Suppose that you build a trapezoidal map of the segments, but whenever two segments intersect, there are two bullet paths shot (up and down) from such a point. As a function of n and m, what is the maximum number of trapezoids in the final trapezoidal map? Explain briefly. (Give an exact, not asymptotic, answer.)

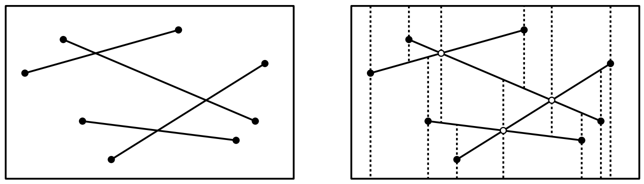


Figure 2: Trapezoidal map with intersections.

(f) 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).

(g) 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?

(h) 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.)

(i) In Fortune's algorithm, suppose that three points p_i, p_j, and p_k are involved in a vertex event (also known as a circle event). Let $c = (c_x, c_y)$ denote the center of their circumcircle, and let r denote its radius. What is the y-value where the vertex event is processed?

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

(i) The Euclidean minimum spanning tree of P is a subgraph of the DT.

(ii) Among all triangulations of P, the DT maximizes the minimum angle.

(iii) Among all triangulations of P, the DT minimizes the maximum angle.

(iv) Among all triangulations of P, the DT minimizes the total sum of edge lengths.

(v) The DT is a t-spanner, for some constant t.

Problem 2. For this problem give an exact bound for full credit and an asymptotic (big-Oh) bound for partial credit. Assume general position.

(a) You are given a convex polygon P in the plane having n_P sides and an x-monotone polygonal chain Q having n_Q sides (see Fig. 3(a)). What is the maximum number of intersections that might occur between the edges of these two polygons?

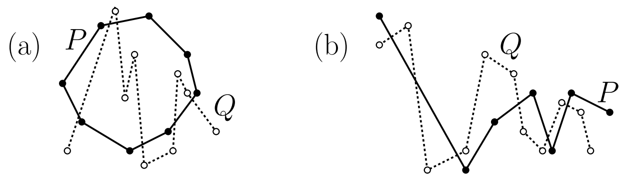


Figure 3: Maximum number of intersections.

(b) Same as (a), but P and Q are both polygonal chains that are monotone with respect to the x-axis (see Fig. 3(b)).

(c) Same as (b), but P and Q are both monotone polygonal chains, but they may be monotone with respect to two different directions.

Problem 3. A simple polygon P is *star-shaped* if there is a point q in the interior of P such that for each point p on the boundary of P, the open line segment $\overline{qp}$ lies entirely within the interior of P (see Fig. 4). Suppose that P is given as a counterclockwise sequence of its vertices $\langle v_1, v_2, \ldots, v_n \rangle$. Show that it is possible to determine whether P is star-shaped in $O(n)$ time. (Note: You are *not* given the point q.) Prove the correctness of your algorithm.

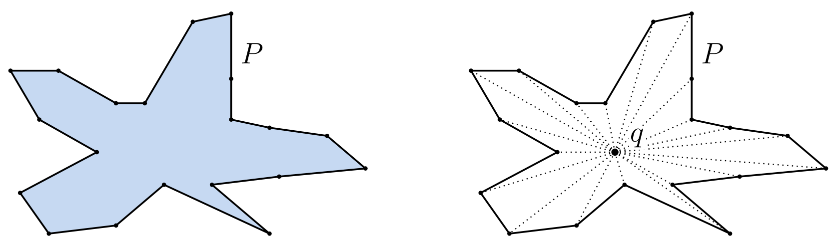


Figure 4: Determining whether a polygon is star-shaped.

Problem 4. 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)).

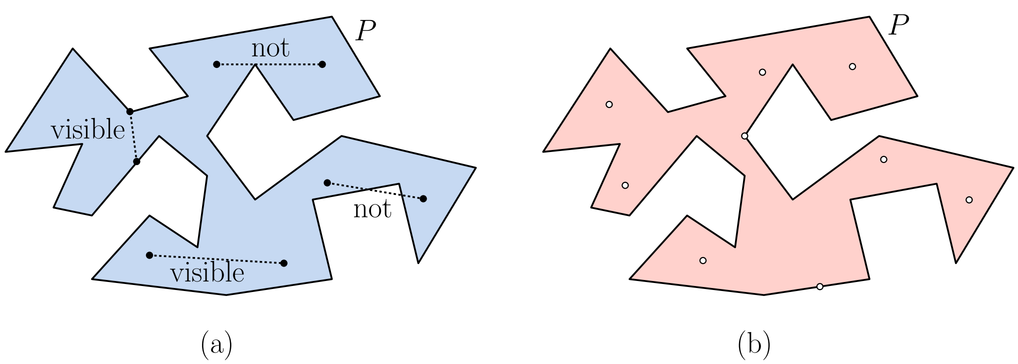


Figure 5: (a) Visibility and (b) a guarding set of size 9 for P.

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.

Problem 5. A *slab* is the region lying between two parallel lines. You are given a set of n slabs, where each is of vertical width 1 (see Fig. 6). Define the *depth* of a point to be the number of slabs that contain it. The objective is to determine the maximum depth of the slabs using plane sweep. (For example, in Fig. 6 the maximum depth is 3, as realized by the small triangular face in the middle.)

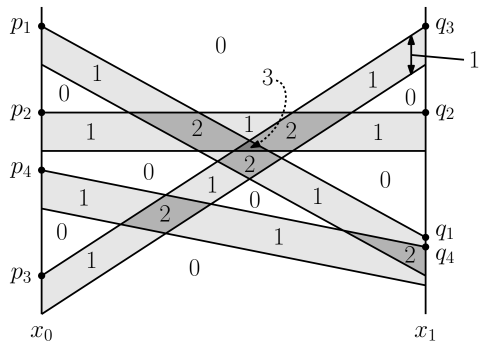


Figure 6: Maximum depth in a set of slabs.

We assume that the slabs lie between two parallel lines at $x = x_0$ and $x = x_1$. The ith slab is identified by the segment $\overline{p_i q_i}$ that forms its upper side (and the lower side is one unit below this). Let I denote the number of intersections between the line segments (both upper and lower) that bound the slabs. Present an $O((n + m) \log n)$-time algorithm to determine the maximum depth. (Hint: Use plane-sweep.)

Problem 6. 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 x_i, its upper y-coordinate y_i^+ and its lower y-coordinate y_i^-. A *transversal* is defined to be a line $\ell : y = ax + b$ that intersects all of these segments (see Fig. 7).

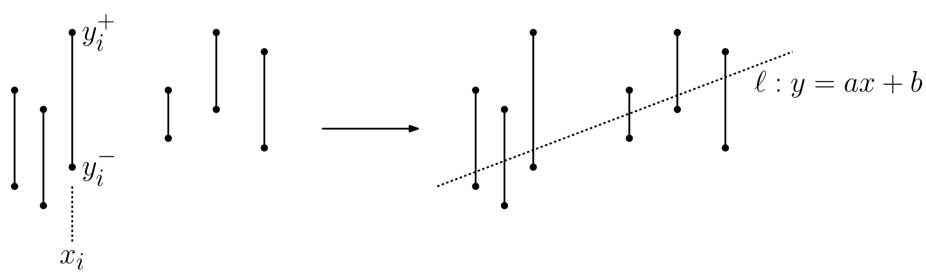


Figure 7: Existence of a transversal.

Present an efficient an algorithm, which given S determines whether there exists a transversal. (Hint: $O(n)$ time is possible.) Justify your algorithm's correctness and derive its running time.

Problem 7. You are given two sets of points in the plane, the red set R containing n_r points and the blue set B containing n_b points. The total number of points in both sets is $n = n_r + n_b$. Give an $O(n)$ time algorithm to determine whether the convex hull of the red set intersects the convex hull of the blue set. If one hull is nested within the other, then we consider them to intersect. (Hint: It may be easier to consider the question in its inverse form, do the convex hulls *not* intersect.)

Problem 8. Given a set of n points P in the plane, we define a subdivision of the plane into rectangular regions by the following rule. We assume that all the points are contained within a bounding rectangle. Imagine that the points are sorted in increasing order of y-coordinate. For each point in this order, shoot a bullet to the left, to the right and up until it hits an existing segment, and then add these three bullet-path segments to the subdivision (see Fig. 8(a)).

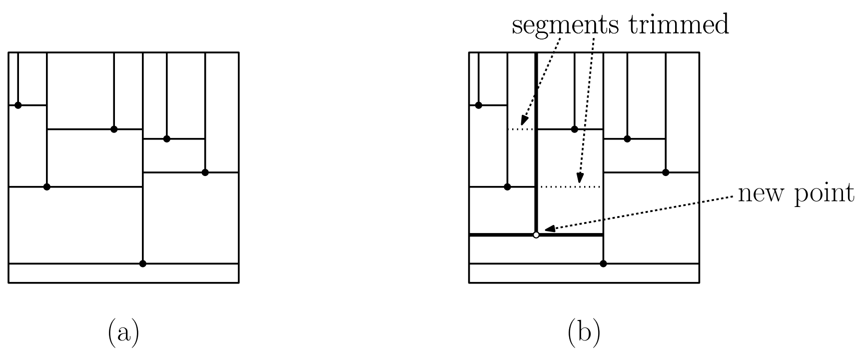


Figure 8: Building a subdivision.

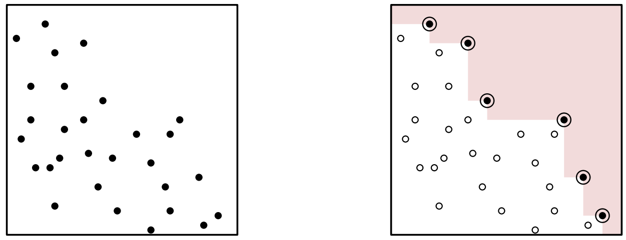
(a) Show that the resulting subdivision has size $O(n)$ (including vertices, edges, and faces).

(b) Describe an algorithm to add a new point to the subdivision and restore the proper subdivision structure. Note that the new point may have an arbitrary y-coordinate, but the subdivision must be updated as if the points had been inserted in increasing order of y-coordinate (see Fig. 8(b)).

(c) Prove that if the points are added in random order, then the expected number of structural changes to the subdivision with each insertion is $O(1)$.

Problem 9. Given two points $p_1 = (x_1, y_1)$ and $p_2 = (x_2, y_2)$ in the plane, we say that p_2 *dominates* p_1 if $x_1 \leq x_2$ and $y_1 \leq y_2$. Given a set of points $P = \{p_1, p_2, \ldots, p_n\}$, a point p_i is said to be *Pareto maximal* if it is not dominated by any other point of P (shown as black points in Fig. 9(b)).

Suppose further that the points of P have been generated by a random process, where the x-coordinate and y-coordinate of each point are independently generated random real numbers in the interval $[0, 1]$.



(a) (b)

Figure 9: Paresto maxima.

(a) Assume that the points of P are sorted in increasing order of their x-coordinates. As a function of n and i, what is the probability that p_i is maximal? (Hint: Consider the points p_j, where $j \geq i$.)

(b) Prove that the expected number of maximal points in P is $O(\log n)$.

Problem 10. Consider an n-sided simple polygon P in the plane. Let us suppose that the leftmost edge of P is vertical (see Fig. 10(a)). Let e denote this edge. Explain how to construct a data structure to answer the following queries in $O(\log n)$ time with $O(n)$ space. Given a ray r whose origin lies on e and which is directed into the interior of P, find the first edge of P that this ray hits. For example, in the figure below the query for ray r should report edge f. (Hint: Reduce this to a point location query in an appropriate planar subdivision.)

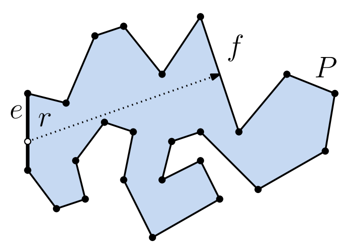


Figure 10: Ray-shooting queries.

Problem 11. You are given a set P of n points in $\mathbb{R}^2$. 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.)

(a) The input to the query is a nonvertical line ℓ. Such a line 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. 11(a)). The answer is the maximum vertical distance h between two lines parallel to h that lie between $P^+(\ell)$ and $P^-(\ell)$ (see Fig. 11(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 consists of the two lines ℓ^- and ℓ^+ of minimum and maximum slope, respectively, that separate

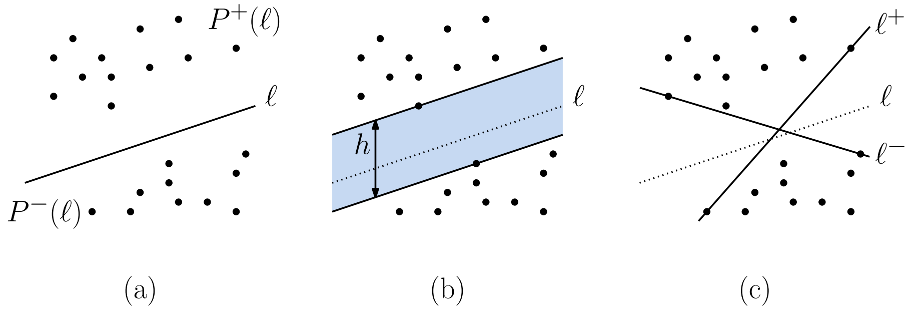


Figure 11: Separation queries.

$P^+(\ell)$ from $P^-(\ell)$ (see Fig. 11(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 12. 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. 12.)

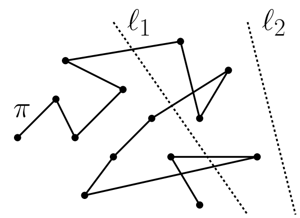


Figure 12: Path crossing queries.

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. 12 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 13. 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 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. 13(a)). The edge (p, q) is in the box graph if and only if box(p, q) contains no other point of P (see Fig. 13(b)). Show that the box graph is a subgraph of the Delaunay triangulation of P.

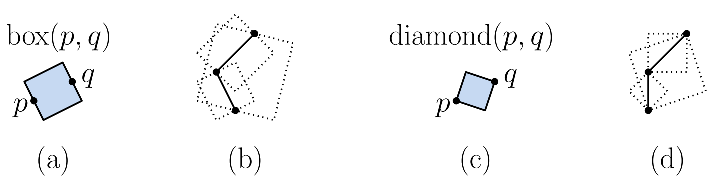


Figure 13: The box and diamond graphs.

(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. 13(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. 13(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.)

Problem 14. You are given a set of n sites P in the plane. Each site of P is the center of a circular disk of radius 1. The points within each disk are said to be *safe*. We say that P is *safely connected* if, given any $p, q \in P$, it is possible to travel from p to q by a path that travels only in the safe region. (For example, the disks of Fig. 14(a) are connected, but the disks of Fig. 14(b) are not.)

Present an $O(n \log n)$ time algorithm to determine whether such a set of sites P is safely connected. Justify the correctness of your algorithm and derive its running time.

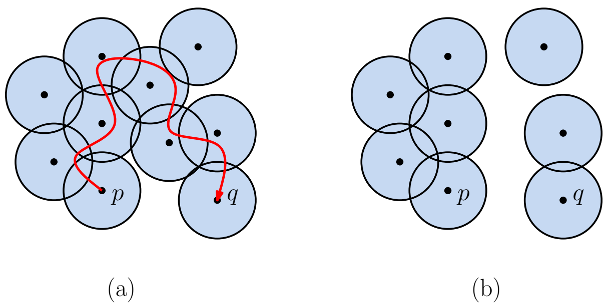


Figure 14: Safe connectivity.

Problem 15. In class we argued that the number of parabolic arcs along the beach line in Fortune's algorithm is at most $2n - 1$. The goal of this problem is to prove this result in a somewhat more general setting.

Consider the beach line at some stage of the computation, and let $\{p_1, \ldots, p_n\}$ denote the sites that have been processed up to this point in time. Label each arc of the beach line with its the associated site. Reading the labels from left to right defines a string. (In Fig. 15 below the string would be "$p_2p_1p_2p_5p_7p_9p_{10}$".)

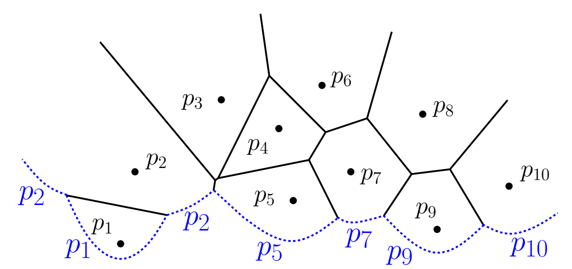


Figure 15: Beach-line complexity.

(a) Prove that for any i, j, the following alternating subsequence *cannot* appear anywhere within such a string:

$$\ldots p_i \ldots p_j \ldots p_i \ldots p_j \ldots$$

(b) Prove that any string of n distinct symbols that does not contain any repeated symbols $(\ldots p_i p_i \ldots)$ and does not contain the alternating sequence[1] of the type given in part (a) cannot be of length greater than $2n - 1$. (Hint: Use induction on n.)

[1]Sequences that contain no forbidden subsequence of alternating symbols are famous in combinatorics. They are known as *Davenport-Schinzel sequences.* They have numerous applications in computational geometry, this being one.