Homework 2

Due by the start of class on Tuesday, Feb 17. (Submissions will be through Gradescope.) Late homeworks are not accepted (unless an extension has been prearranged) so please turn in whatever you have completed by the due date. Unless otherwise specified, you may assume that all inputs are given in *general position*.

Problem 1.

(a) What is the maximum number of intersections possible between two x-monotone polygonal curves having n and m edges, respectively (see Fig. 1). You should assume that the curves are in general position assumption, implying that there are no duplicate x-coordinates or parallel edges. Express your answer as an exact (not asymptotic) function of n and m, and justify your answer. (For full credit, your answer should be correct up to a constant additive term. It is okay if your bound only works for even and/or odd values of n and m.)

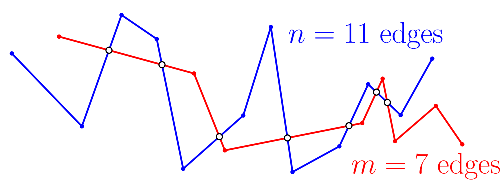

Figure 1: Number of intersections.

(b) (Optional–Ungraded) Present a generic example showing that your bound is tight. (By "generic," we mean that it should be obvious how to extend your example to work for arbitrarily large values of n and m.)

Problem 2. Given a simple polygon P with n vertices, recall that the addition of any diagonal (an internal line segment joining two visible vertices of P) splits P into two simple polygons with n_1 and n_2 vertices respectively, where $n_1 + n_2 = n + 2$.

(a) Prove that given any simple polygon P with $n \geq 4$, there exists a *balanced diagonal*, meaning that splits P such that $\min(n_1, n_2) \geq \lfloor n/3 \rfloor$. (**Hint:** Show that such a diagonal exists as an edge of *any* triangulation of P.)

(b) (Optional–Ungraded) Show that the constant 1/3 is the best possible, in that for any $c > 1/3$, there exists a polygon such that *any* diagonal chosen results in a split such that $\min(n_1, n_2) < cn$. (You can provide a generic drawing to illustrate your example. Your example does not need to work for all values of n, as long as it holds for arbitrarily large values of n.) **Hint:** Create a polygon that consists of three identical "spokes" joined around a tight central hub, such that there is no diagonal that runs from the interior of one spoke to another.

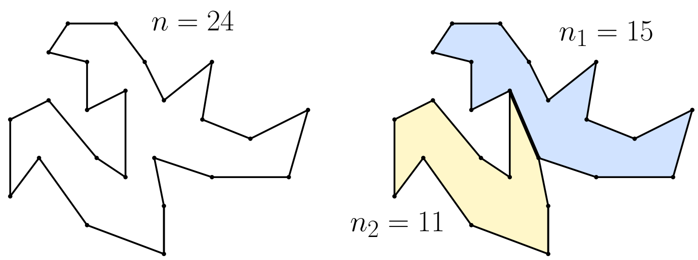

Figure 2: Balanced diagonal.

Problem 3. An important application of computational geometry in robotics is computing shortest paths. In this problem, we will consider a simple instance, which can be solved by plane sweep.

You are given a set $S = \{s_1, \ldots, s_n\}$ of vertical line segment obstacles in $\mathbb{R}^d$, where segment s_i has upper endpoint a_i and lower endpoint b_i. You are also given a start point s and destination point t, where $s_x < t_x$ (see Fig. 3(a)). Define a *rectilinear* (s,t)*-path* to be a polygonal curve that starts at s, ends at t, and all its edges are either horizontal or vertical. Such a path is *obstacle free* if none of its horizontal segments intersects the interior of any obstacle segment. The *length* of such a path is the sum of its edge lengths. The *shortest rectilinear path problem* is to compute an obstacle-free rectilinear (s,t)-path of minimum length (see Fig. 3(b)). (As always, you may assume that the input is in general position, meaning no duplicate x- or y-coordinates.)

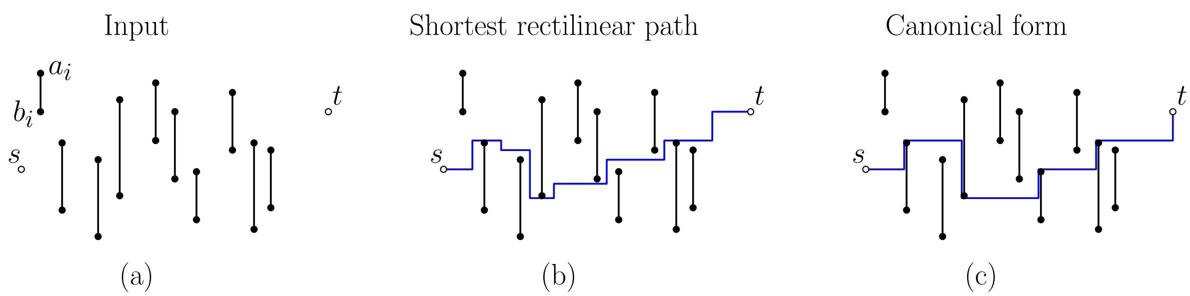

Figure 3: Shortest rectilinear path problem.

(a) Prove that any rectilinear shortest path is x-monotone, in the sense that all of its horizontal segments are directed in the positive x-direction.

(b) An (s,t)-rectilinear path is said to be in *canonical form* if it is x-monotone, and other than along the vertical lines through s and t, its vertices lie on the obstacle segments (either in the obstacle interior or at an obstacle vertex). Prove that for any valid input, there exists a shortest rectilinear (s,t)-path that is in canonical form (see Fig. 3(c)).

(c) Present an efficient algorithm that solves the shortest rectilinear path problem for a given input. (Hint: Use plane sweep, where the sweep-line status stores the endpoints of all valid canonical paths ending at the sweep line and the information needed to compute their lengths efficiently. Aim for a running time of $O(n \log n)$.)

Problem 4. (Optional–Ungraded) Suppose that you are given m convex polygons $P_1, \ldots, P_m$ in the plane (see Fig. 4(a)). Let n_i denote the number of vertices on P_i, and let $\langle v_{i,1}, \ldots, v_{i,n_i} \rangle$ denote the vertices of this polygon listed in counterclockwise order, starting at the leftmost vertex of P_i (that is, the one with the smallest x-coordinate). Two polygons P_i and P_j are said to *intersect* if they contain any point in common (that is, either their boundaries intersect or one polygon is contained within the other).

(a) Present an efficient algorithm that determines whether the *boundaries* of any two polygons of the set intersect (see Fig. 4(b)). (The output is either "yes" or "no"; you do not need to report all the intersections.) Let $n = \sum_{i=1}^{m} n_i$ denote the total number of vertices. For full credit, you algorithm should run in $O(n \log m)$ time.

(b) Explain how to modify your answer to (a) to also detect intersections where one polygon is *containined* in the interior of another (see Fig. 4(c)).

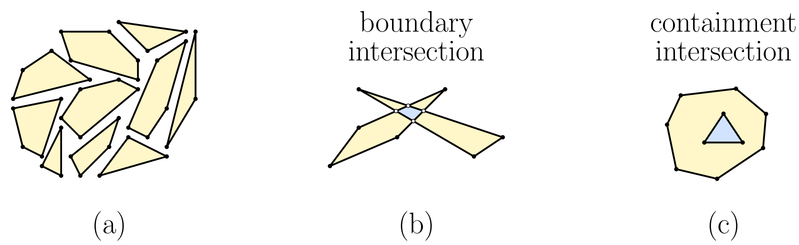

Figure 4: (a) Nonintersecting convex polygons, (b) boundary intersection, (c) containment intersection.

Note: Challenge problems are not graded as part of the homework. The grades are recorded separately. After final grades have been computed, I may "bump-up" a grade that is slightly below a cutoff threshold based on these extra points. (But there is no formal rule for this.)

Challenge Problem: Present an efficient algorithm which, given a set $P = \{p_1, \ldots, p_n\}$ of n points on the integer grid, computes the polygon of minimum perimeter that encloses these points, subject to the condition that the sides of this enclosure can be horizontal, vertical, or sloped at $\pm 45°$ (see Fig. 5). Prove that the enclosure produced by your algorithm has the minimum perimeter. $O(n)$ time is possible, but $O(n \log n)$ time acceptable. (**Hint:** There are generally many enclosures with the same perimeter. Show that there exists a minimum-perimeter enclosure of a particularly simple structure.)

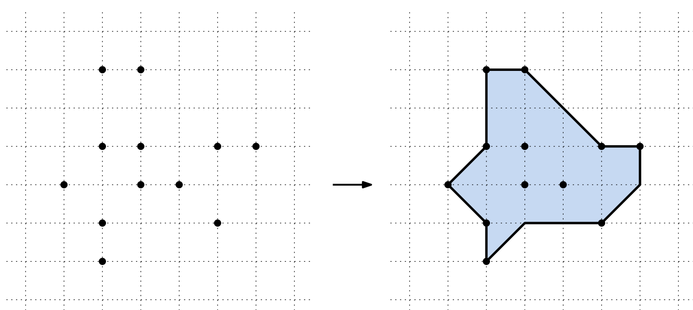

Figure 5: Minimum perimeter enclosure.