CMSC 754: Fall 2021 Dave Mount

Homework 2: Duality, Linear Programming, and Point Location

Handed out Thursday, Oct 14. Due: **9:30am, Tuesday, Oct 26** (submission through Gradescope as with Homework 1). No late homeworks will be accepted, 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.* Also, when asked to give an algorithm with running time $O(f(n))$, it is allowed to give a *randomized* algorithm with *expected* running time $O(f(n))$.

Problem 1. Consider the two segments $s_1 = \overline{p_1t_1}$ and $s_2 = \overline{p_2t_2}$ shown in Fig. 1.

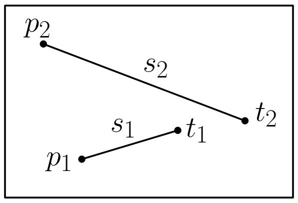

Figure 1: Problem 1: Trapezoidal map and point location.

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

(b) Show the *point-location data structure* resulting from the construction given in class, assuming the insertion order $\langle s_1, s_2\rangle$. (We will give partial credit if your data structure works correctly, even though it does not match the construction given in class.)

Please follow the convention given in class for the node structure. (In particular, for y-nodes, the left (resp., right) child corresponds to the region above (resp., below) the segment.)

Problem 2. Euler's formula is useful for computing the combinatorial properties of planar subdivisions. A planar graph (or more accurately, a cell complex) is a subdivision of the plane into vertices (0-dimensional), edges (1-dimensional), and faces (2-dimensional). Let v, e, and f denote the number of vertices, edges, and faces, respectively, in a given cell complex. (Note that f includes the unbounded face that extends to infinity.) Euler's formula states that these quantities are related as

$$2 = v - e + f$$

For example, in the Fig. 2(a) we show a triangulation of a set of $v = 16$ vertices with $h = 10$ vertices on the convex hull. In Fig. 2(b) we show a quadrilateral cell complex. Using Euler's formula, answer each of the following questions.

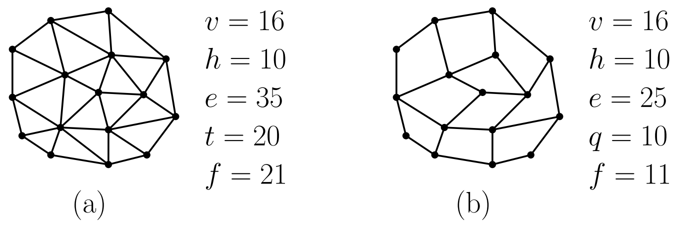

Figure 2: Problem 2: Applications of Euler's formula.

(a) Given a triangulation with v vertices, where h of these lie on the convex hull, use Euler's formula to derive the formula for the number of triangles t and the number of edges e in the triangulation as functions of v and h. (Hints: Observe that $3t = 2e - h$. This follows because $3t$ counts the total number of edges incident to all the triangles, and this counts every edge twice except the h edges that form the convex hull, which are only counted once. Also observe that the number of faces is $f = t + 1$, because the infinite exterior face is counted as a face.)

(b) Given a quadrangulation (a cell complex where each face has four edges, excluding the face lying outside the convex hull) with v vertices, where h of these lie on the convex hull, use Euler's formula to derive the formula for the number of quadrangles q and the number of edges e in the quadrangulation as functions of v and h.

(c) Explain why your answer to (b) implies that a quadrilateralization does not exist if the number of hull vertices is odd.

Problem 3.

(a) You are given two sets of points, red and blue, in the plane. Let $R = \{r_1, \ldots, r_n\}$ be the red points and $B = \{b_1, \ldots, b_n\}$ be the blue points. The problem is to determine a pair of parallel, nonvertical lines ℓ_R and ℓ_B such that all the points of R lie on or above ℓ_R, all the points of B lie on or below ℓ_B, and the signed vertical distance from ℓ_B to ℓ_R is as large as possible. (More formally, if y_B and y_R are the y-intercepts of these lines, we want to maximize $y_R - y_B$.)

Note that if ℓ_R lies above ℓ_B, this distance is positive and if (as shown in the figure below) ℓ_R is below ℓ_B, this distance is negative. (When negative, the objective is to minmize the absolute value of the distance.) Present an $O(n)$ time algorithm to solve this problem. (**Hint:** Reduce to linear programming.)

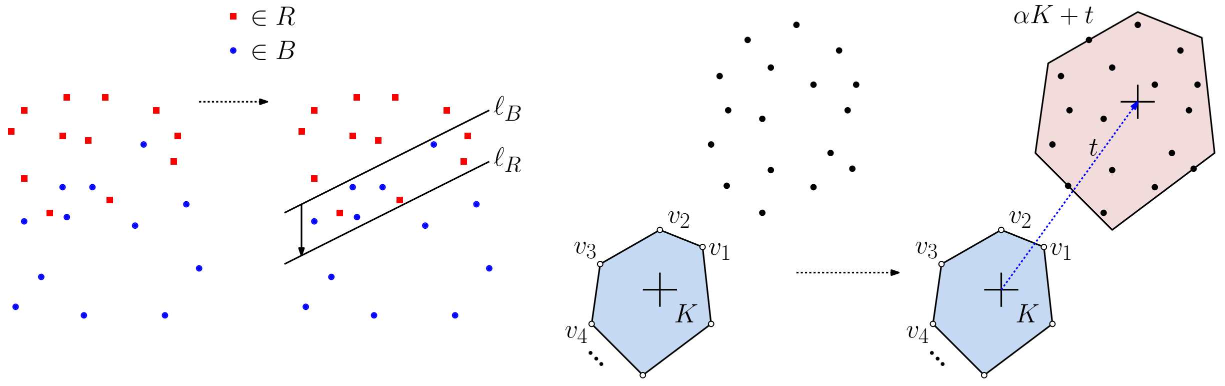

Figure 3: Problem 3: (a) Separating point sets and (b) enclosing points by a polygonal shape.

(b) You are given a convex polygon K in $\mathbb{R}^2$, presented by its vertices in counterclockwise order $\langle v_1, \ldots, v_k \rangle$. We assume that the origin is contained in K's interior (see Fig. 3(b)). Given a positive real scalar α and a translation vector $t = (t_x, t_y)$, let $t + \alpha K$ denote the convex polygon that arises by scaling all the vertices of K uniformly by a factor of α (about the origin) and then translating them by vector t (see Fig. 3(c)). Such a body is called a *homothet* of K of scale α.

Present an algorithm, which given a convex polygon K (as described above) and an n-element point set $P = \{p_1, \ldots, p_n\}$ in $\mathbb{R}^2$, computes the homothet of K of smallest (positive) scale that contains all the points of P. This can be solved in $O(kn)$ time.

Hint: The reduction to LP involves multiple steps. Here are some suggestions:

(i) Explain how to express K as the intersection of k halfplanes $\{h_1, \ldots, h_k\}$.

(ii) Consider any halfplane $h = \{(x, y) \mid ax + by \leq c\}$, and let $h' = t + \alpha h$ denote the halfplane that arises by scaling the points of h by α (about the origin) and translating by the vector t. Given a point $p = (p_x, p_y)$. Prove that $p \in h'$, if and only if

$$a\frac{p_x - t_x}{\alpha} + b\frac{p_y - t_y}{\alpha} \ \leq \ c.$$

(iii) Use (i) and (ii) to obtain an $O(kn)$ time algorithm that computes the minimum-scale homothet of K enclosing P.

Since we have not done any examples of linear programming applications in class, here is simple example of how to answer one of these problems.

Sample Problem: Present an $O(n)$ time algorithm, which given two sets of points $R = \{r_1, \ldots, r_n\}$ and $B = \{b_1, \ldots, b_n\}$, both in $\mathbb{R}^3$, determines whether their exists a plane h in $\mathbb{R}^3$ such that all the points of R lie on or above h and all the points of B lie on or below h.

Sample solution: We reduce the problem to linear programming in $\mathbb{R}^3$. Let's assume that each $r_i \in R$ is given in coordinate form as $(r_{i,x}, r_{i,y}, r_{i,z})$ and similarly for B. Let's model h by the equation $z = ax + dy + e$, for some real parameters a, d, and e. To enforce the condition that each r_i lies on or above h and each b_j lies on or below it, we add the constraints

$$\begin{aligned} r_{i,z} &\geq ar_{i,x} + dr_{i,y} + e, && \text{for } 1 \leq i \leq n \\ b_{j,z} &\leq ab_{j,x} + db_{j,y} + e, && \text{for } 1 \leq j \leq n. \end{aligned}$$

We then invoke LP with $2n$ constraints in $\mathbb{R}^3$ (with the variables (a, d, e)). Since this is a yes-no answer, we don't really care about the objective function. We can set it arbitrarily, for example, "maximize e" (which is equivalent to using the objective vector $c = (0, 0, 1)$).

We interpret the LP's result as follows. If the result is "infeasible", then we know that no such plane exists. If the answer is "feasible" or "unbounded", then we assert that such a plane exists (assuming general position). This is clearly true if the result is "feasible", since we can just take h to be the plane associated with the optimum vertex (a, d, e). If the result is "unbounded", then the plane is vertical, but there exists a perturbation such that R lies above and B lies below.

Problem 4. The objective of this problem is to explore some interesting properties of trapezoidal maps (which apply more generally to many geometric structures). Throughout this problem, $S = \{s_1, \ldots, s_n\}$ denotes a set of n nonintersecting, nonvertical line segments in the plane. Let $\mathcal{T}(S)$ denote the trapezoidal map of these segments. We say that a trapezoid $\Delta \in \mathcal{T}(S)$ is *incident* on a segment $s \in S$ if s borders Δ from above or below, or if one of s's endpoints bounds Δ from the left or the right (see Fig. 2(a)). For $s \in S$, define $\deg(s)$ to be the number of trapezoids of $\mathcal{T}(S)$ that are incident on s (in Fig. 4(a), $\deg(s) = 7$).

(a) Given any set S of n segments, prove that there exists a constant c, such that, for all sufficiently large n, $\sum_{s \in S} \deg(s) \leq cn$.

(b) Let c be the constant derived in your solution to (a). We say that a segment $s \in S$ is *long* if $\deg(s) \geq 2c$, and otherwise we say that s is *short*. Let $S' \subseteq S$ be the set of short segments of S. Prove that there exists a constant c' (which may depend on c) such that, for all sufficiently large n, $|S'| \geq n/c'$.

(c) We say that two segments $s_i, s_j \in S$ are *adjacent* if there exists a trapezoid $\Delta \in \mathcal{T}(S)$ that is incident on both s_i and s_j. Define an *independent set* of S to be a subset of S whose elements are pairwise non-adjacent (see Fig. 4(b)). Given the previous constants c and c', prove that there

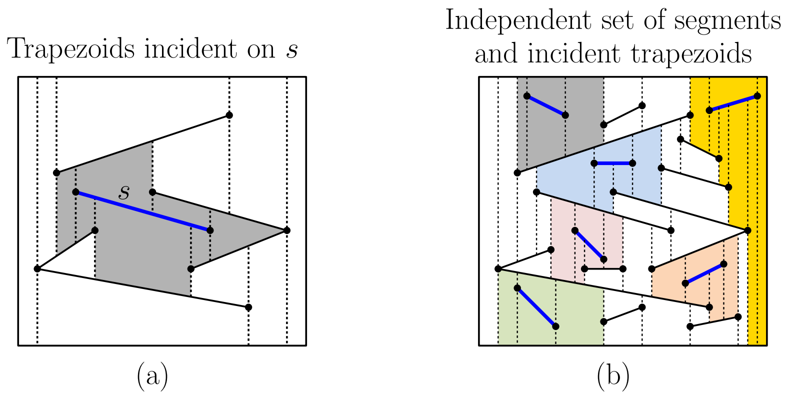
Trapezoids incident on s
s
(a)
Independent set of segments and incident trapezoids
(b)

Figure 4: Problem 4: Independent sets in a trapezoidal map.

exists a constant $c'' > 1$ (depending on c and c') such that, for all sufficiently large n, S contains an independent set of size at least n/c'' *consisting entirely of short segments*. (**Hint:** Use a greedy approach.)

You might wonder why we care about independent sets at all. The existence of large independent sets is of critical to the efficiency of many algorithms based on divide-and-conquer. The idea is to find a large independent set, remove it, and solve the problem recursively on the remaining objects. (Since the number of remaining objects decreases by a constant factor, these total time for all these recursive calls will be small.) When the recursion returns, add back the elements of the independent set, and solve the problem. Since the various pieces are independent of each other, the solutions of these independent subproblems will not interact with each other.

Problem 5. You are given two vertical lines at $x = 0$ and $x = 1$ and a set of n (nonvertical) line segments, $s_i = \overline{a_i b_i}$. The left endpoint a_i of each segment lies on a vertical line $x = 0$ and the right endpoint b_i lies on the vertical line $x = 1$ (see Fig. 5(a)). Scanning from left to right, whenever two segments intersect, the segment with the lower slope "terminates" and the one with the higher slope continues on (see Fig. 5(b)). Let us also add an imaginary "sentinel segment" s_0 that runs along the right vertical line.

Observe that for $1 \leq i \leq n$, every segment s_i is *terminated* by some other segment. If the segment survives to the right side, then it is terminated by segment 0. (For example, in Fig. 5(b), segment 1 is terminated by segment 2, segments 2, 3, and 4 are all terminated by segment 5, segment 5 is terminated by ∞, and so on.)

(a) Assuming that the segments are given in sorted order according to their left endpoints (say, from top to bottom as shown in our figure), present an efficient algorithm that determines the n-element list of the indices of the segment terminators. (For example, for the input shown in the figure, the output would be $\langle 2, 5, 5, 5, 0, 7, 8, 0 \rangle$.) **Hint:** $O(n)$ time is possible.

(b) Suppose that we instead insert the segments in random order. A new segment $s = \overline{ab}$ runs from left to right until it is terminated by the first segment of higher slope that it intersects. In addition all the segments from the existing structure of lower slope that intersect s are now terminated by s. (For example, the blue segment s in Fig. 5(c) and (d) changes the termination points of three existing segments, as indicated by the red arrows.)

Prove that there exists a constant c such that, if the segments are inserted in random order, the expected number of existing segments that change their termination point is at most c. (**Hint:** Apply a backwards analysis.)

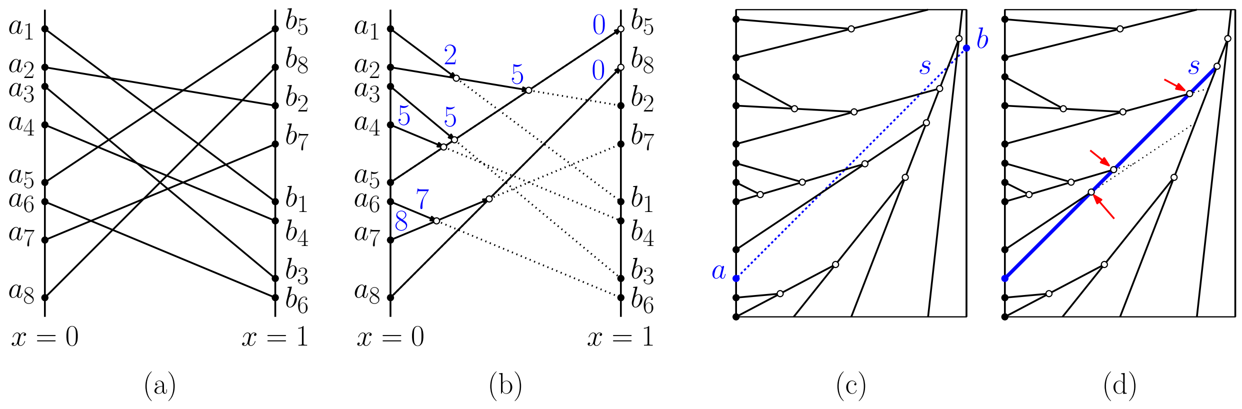

Figure 5: Problem 5(a) and (b): Terminating segments

(c) Recall that in Graham's scan, we computed the upper hull of a set of points by adding the points in left-to-right order. Whenever a point p_i was added, we determined the point p_j of tangency with respect to the current upper hull, and we added the edge $\overline{p_i p_j}$. We removed all points that were "shadowed" by the newly added edge. If you look at the entire history of edges generated by Graham's scan, you obtain a tree-like structure, as shown in Fig. 6 below.

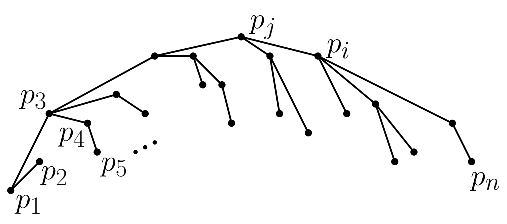

Figure 6: Problem 5(c): Graham's scan history

Prove that there is a equivalency between the tree-like structures of Fig. 5(b) and Graham-scan structure from Fig. 6. In particular, given any set of points $P = \{p_1, \ldots, p_n\}$ in the plane, explain how to map these to a set of segments $S = \{s_1, \ldots, s_n\}$ such that the edge $\overline{p_i p_j}$ is added by Graham's scan if and only if segment s_i is terminated by segment s_j.

Hint: This will involve some form of point-line duality, but you may need to modify the duality transformation given in class. It may also be necessary to change the x-coordinates associated with the left and right sides of the vertical band in the segment-termination problem, even so far as to take the limit as they tend to infinity.

Problem 6. You are given a collection of n nonintersecting circular disks in the plane, each of unit radius. Let $P = \{p_1, \ldots, p_n\}$ denote their center points. Preprocess these disks into a data structure to answer the following queries. Given a unit disk q_i (designated by its center point), determine whether it is possible for this disk to *escape* from the others, meaning that it is possible to move this disk arbitrarily far away from the others without intersecting or moving any of the disks of P.

For example, given the disks in Fig. 7, the disk q_1 can escape from the others, while q_2 cannot. Present an $O(n \log n)$ that constructs such a data structure. Your data structure should have space $O(n)$ and should be able to answer queries in time $O(\log n)$. (**Hint:** Use Voronoi diagrams and point location.)

Challenge Problem. (Challenge problems count for extra credit points. These additional points are factored in only after the final cutoffs have been set, and can only increase your final grade.)

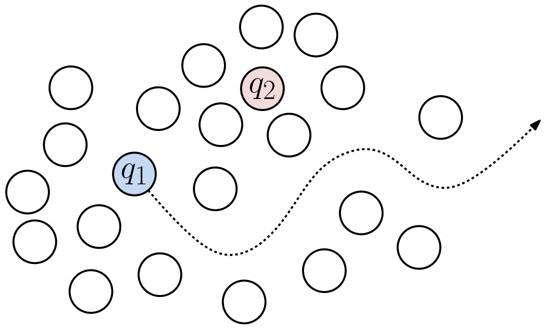

Figure 7: Problem 6. Disk q_1 can escape while q_2 cannot.

Present a randomized incremental algorithm for structure described in Problem 5(b). (In 5(b) you were asked just to show that the expected number of changes is $O(1)$. Here you must compute those changes and update the structure.)

Assume that your input is given as a collection of segments $S = \{s_1, \ldots, s_n\}$, where the endpoints lie on the vertical lines $x = 0$ and $x = 1$. Randomly permute segments and insert them one-by-one into the tree structure described in the problem. The aim is to produce the final structure in expected time $O(n \log n)$.

You may create any additional auxiliary structures you like in order to help achieve the desired running time.

Prove the correctness of your algorithm and derive its expected running time.