Solutions to Quiz 2

Solution 1: The final trapezoidal map and point-location data structure are shown in Fig. 1.

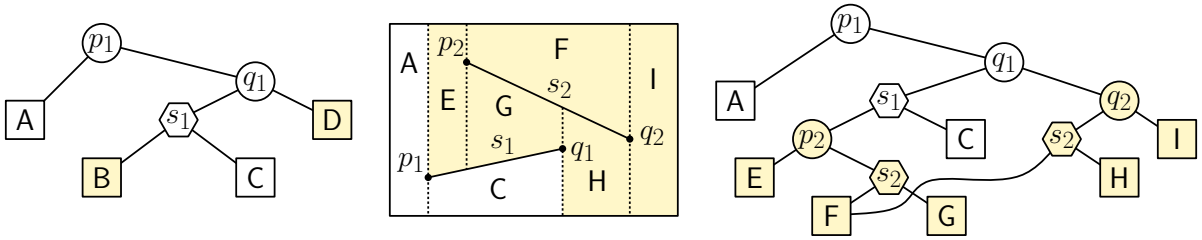

Figure 1: Trapezoidal map and point location.

The construction given in class prioritizes first testing the left endpoint of the segment (if it overlaps the region), then testing the right endpoint (if it overlaps the region), and finally the segment. If multiple search paths lead to the same trapezoid (as with trapezoid F) they share a common node in the tree.

Solution 2:

(a) We will show that the number of triangles is $2v - h - 2$.

First, observe that $3t$ counts each internal edge twice and each edge of the h edges of the convex hull once. Therefore, we have $3t = 2e - h$, or equivalently, $e = (3t + h)/2$. By Euler's formula and the fact that $f = t + 1$, we have

$$\begin{aligned} 2 &= v - e + f = v - \frac{3t+h}{2} + (t+1) \\ \implies t &= 2v - h - 2, \end{aligned}$$

as desired.

(b) We will show that the number of internal edges is $3v - 2h - 3$.

By applying our formula from (a), we have $f = t + 1 = (2v - h - 2) + 1 = 2v - h - 1$. By applying Euler's formula again, we have

$$\begin{aligned} 2 &= v - e + f = v - e + (2v - h - 1) \\ \implies e &= 3v - h - 3. \end{aligned}$$

This counts all the edges. From this, we subtract the h edges that are on the hull, to obtain a final count of $3v - 2h - 3$.

Solution 3:

(a) Let d denote the height along the y-axis where the arrow is shot and let s denote the slope of the arrow's trajectory line. This defines the line equation $\ell : y = sx + d$. The upper/lower endpoints of the ith segment have the y-coordinates $c_{i,y} \pm \frac{h_i}{2}$. We will solve the problem of finding any solution vector $x = (d, s)^\mathsf{T} \in \mathbb{R}^2$ that satisfies the following constraints:

- The shot is made on the positive y-axis, so $d \geq 0$.
- The line passes above all the lower endpoints of all the segments: $sc_{i,x} + d \geq c_{i,y} - \frac{h_i}{2}$, for $1 \leq i \leq n$.
- The line passes below all the upper endpoints of all the segments: $sc_{i,x} + d \leq c_{i,y} + \frac{h_i}{2}$, for $1 \leq i \leq n$.

Since we seek any feasible answer, we do not care about the objective function. Let's say we minimize d, using the objective vector $(-1, 0)$. By the constraint $d \geq 0$, the LP cannot be unbounded. If the LP is feasible, we return pair d and directional vector $u = (1, s)$. Otherwise, the LP is infeasible, and we report that no such line exists. Since we have 2-dimensional LP with $O(n)$ constraints, the running time is $O(n)$ (in expectation).

(b) This can be solved by two invocations of LP in $\mathbb{R}^2$. The constraints will be exactly the same as in (a), but we need to be careful about the objective function. Observe that the line $y = sx + d$ crosses the ith segment at the y-value $p_{i,y} = sc_{i,x} + d$. Therefore, the average displacement of the line from all the centers is

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n}(p_{i,y} - c_{i,y}) &= \frac{1}{n}\sum_{i=1}^{n}p_{i,y} - \frac{1}{n}\sum_{i=1}^{n}c_{i,y} = \frac{1}{n}\sum_{i=1}^{n}(sc_{i,x} + d) - \frac{1}{n}\sum_{i=1}^{n}c_{i,y} \\ &= \left(\frac{1}{n}\sum_{i=1}^{n}c_{i,x}\right)s + d - \left(\frac{1}{n}\sum_{i=1}^{n}c_{i,y}\right) = As + d - B,\end{aligned}$$

where $A = \frac{1}{n}\sum_{i=1}^{n} c_{i,x}$ and $B = \frac{1}{n}\sum_{i=1}^{n} c_{i,y}$. Observe that A and B do not depends on either s or d and can be computed from the inputs. The result is a linear function in s and d. Unfortunately, we need to minimize its absolute value, and this is not a linear function. We make two calls to LP.

- First, we set the objective function to minimize $As + d$ and add the constraint $As + d \geq B$. (Formally, we are to maximize using the objective function given by the vector is $(-A, -1)^\mathsf{T}$.) Note that the LP cannot be unbounded. If it is feasible, we return this answer as the final result, since this is as close as we can get to B from above. If infeasible, we go to the second step.
- Second, we try again, but this time we maximize $As+d$ and add the constraint $As+d \leq B$ (that is, we maximize using the objective function given by the vector is $(A, 1)^\mathsf{T}$). As before, the LP cannot be unbounded. If it is feasible, we return this answer as the final result, since this is as close as we can get to B from below. Otherwise, we conclude that the entire problem is infeasible.

Since we have two instances of 2-dimensional LP with $O(n)$ constraints, the running time is $O(n)$ (in expectation).

Alternatively, we can solve this with a single LP in $\mathbb{R}^3$ through the use of an additional slack variable. (We omit the details.)

Solution 4:

(a) Given the shot description (d, u_x, u_y), let $s = u_y/u_x$ denote the slope of the line carrying the ball. The line equation along which the arrow flies is $\ell : y = sx + d$. Through point-line duality, we can treat this as a point $\ell^* = (s, -d)$ in the dual plane.

Let $A = \{a_1, \ldots, a_n\}$ denote the lower endpoints of the segments, where $a_i = (c_{i,x}, c_{i,y} - \frac{h_i}{2})$. Let $B = \{b_1, \ldots, b_n\}$ denote the upper endpoints, where $b_i = (c_{i,x}, c_{i,y} + \frac{h_i}{2})$ (see Fig. 2). In order for the arrow to hit all the segments, we need ℓ to pass above all the points of A and below all the points of B. By the order-reversing property of duality, this is equivalent to requiring that the dual point ℓ^* lies below all the dual lines A^* and above all the dual lines B^*. We will test these two conditions separately for A and B. If neither is violated, then the arrow pierces all the segments. Otherwise, we determine the leftmost segment that was missed by either of these sets, and return the leftmost of these.

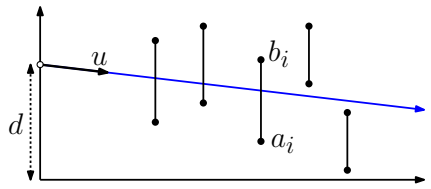

Figure 2: Arrow-shooting queries.

Let's just consider the points of A, and the points B follows by a symmetrical construction. We may assume that the points of A have been sorted from left to right. Consider the dual line $a_i^* : y = a_{i,x}x - a_{i,y}$. Because we want the leftmost segment that is missed (that is, the minimum $a_{i,x}$), our problem reduces to computing the dual line a_i^* having the minimum slope value $(a_{i,x})$ such that ℓ^* lies above a_i^* (see Fig. 3(a)).

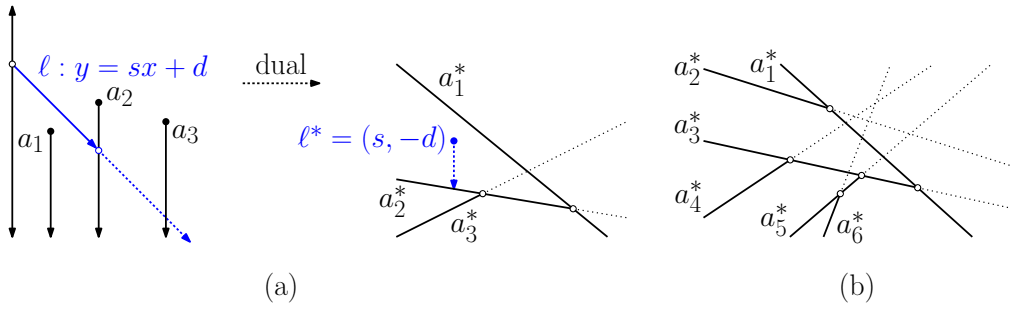

Figure 3: Arrow-shooting queries. (Dual is not geometrically accurate.)

To turn this into a point location problem, we construct a collection of line segments as follows. We consider all the dual lines A^*, tracing each from left to right. Whenever two lines

intersect, we terminate the one with the higher slope and continue along the segment with the lower slope. This results in a tree-like structure of line segments (see Fig. 3(b)). (Although the line "segments" are unbounded, our point location data structure can easily be modified to handle this.)

Each line is trimmed when it hits a line of less slope, so there are n nonintersecting segments that result. (Well, this is not entirely accurate, since right endpoint of one segment may intersect the interior of another. If it bothers you, it is easy to split the second segment at this intersection point. This creates $n-1$ new vertices, and hence the total number of segments is at most $2n-1$.)

The resulting data structure has one segment for each obstacle of A, for an expected size of $O(n)$. Queries are answered in expected query time $O(\log n)$.

(b) The key observation is that the lines that are parallel to ℓ share the same slope, and hence (because the x-coordinate is the slope in the dual) their duals lie on the vertical line through ℓ^*. The desired points a_i and b_j correspond to the dual lines that lie immediately above and below ℓ^*.

We can answer this with a small modification to the answer for (a). When we apply the point location data structure, we can determine which trapezoid contains the dual point ℓ^*. The point-location data structure tells us which trapezoid contains ℓ^*, and in additional $O(1)$ time, we can determine the segments bounding this trapezoid on the top and bottom.

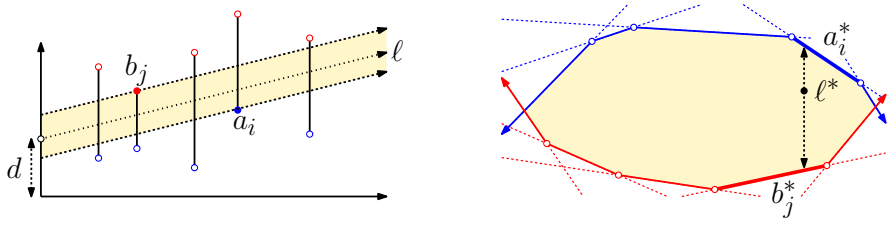

Figure 4: Arrow shooting vertical variability. (Not drawn to scale.)

An alternative, which does not require point location, is to compute the lower envelope of the dual lines A^* and the upper envelope of the dual lines of B^*. Given the dual point ℓ^*, we can employ binary search along the x-coordinates of each of the envelopes to determine the lowest dual line of $a_i^* \in A^*$ (resp., highest dual line of $b_j^* \in B^*$) that lies above (resp., below) ℓ^* (see Fig. 4). The associated primal points a_i and b_i are the highest (resp., lowest) points below (resp., above) ℓ.