CMSC 754: Spring 2026 Dave Mount

Homework 4

Due by the start of class on Tuesday, Mar 10. (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. Consider the trapezoidal map shown in Fig. 1.

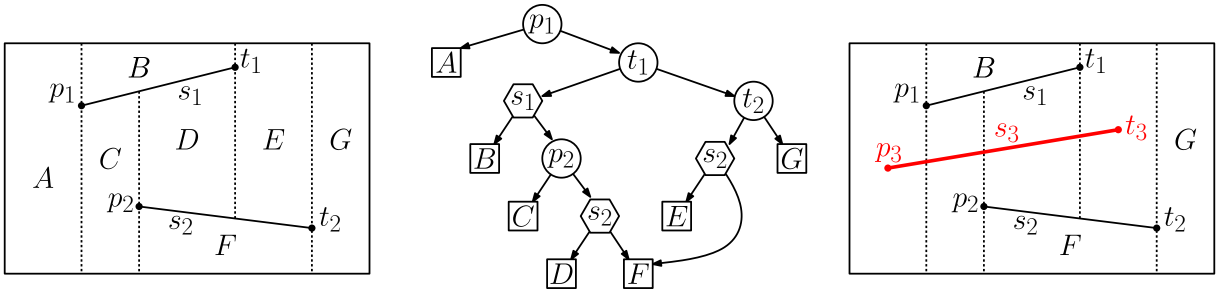


Figure 1: Trapezoidal map insertion and point location data structure.

(a) Show the trapezoidal map that results from inserting the segment $s_3 = \overline{p_3t_3}$. **Suggestion:** To make the grading easier, label the newly created new trapezoids $H, I, J, \ldots$ from left to right.

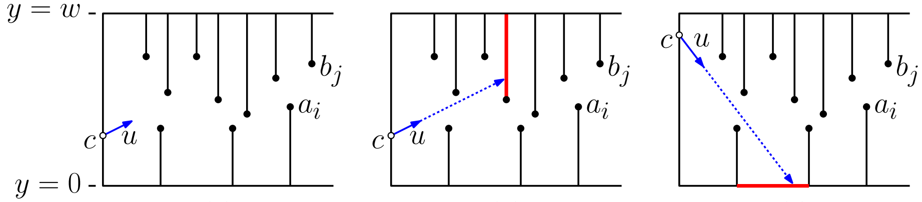
(b) Show the point-location data structure resulting from the insertion of s_3, assuming the construction given in class. **Note:** Be sure to draw internal nodes so that the left and right child pointers are clearly distinguishable. We will give partial credit if you produce a valid point-location structure, even if it is not the same as the one from the algorithm given in class.

(c) In your answer to (b), how many new nodes (both internal and leaves) were created and how many existing nodes (both internal and leaves) were removed?

Problem 2. This problem involves devising a data structure for answering a special type of ray-shooting query. The input consists of a $w \times \ell$ rectangle with origin at the lower-left corner and point sets $A = \{a_1, \ldots, a_m\}$ and $B = \{b_1, \ldots, b_m\}$ all lying within the rectangle, where $a_i = (a_{i,x}, a_{i,y})$ and $b_j = (b_{j,x}, b_{j,y})$. The points of A are the upper endpoints of vertical segments that extend down to the bottom of the rectangle, and the points of B are the lower endpoints of vertical segments that extend up to the top of the rectangle (see Fig. 2(a)).

The ray is determined by two quantities (see Fig. 2(a))

- the y-coordinate c along the left edge of the rectangle where your shot starts, and
- a directional vector $u = (u_x, u_y)$ of a ray along which the shot travels.



(a) (b) (c)

Figure 2: Ray-shooting queries.

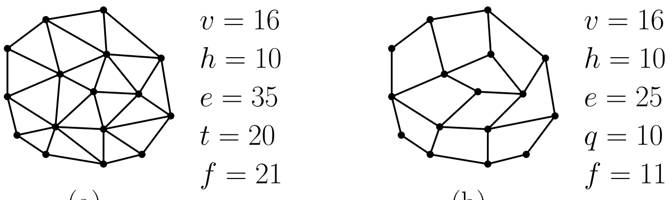
Your objective is to preprocess A and B into a data structure so that given the triple (c, u_x, u_y), you can efficiently determine which segment is hit first (or if no segment is hit, does the ball hit the upper, lower, or right side of the rectangle). You may assume that $0 \leq c \leq w$ and $u_x > 0$. Your data structure should use $O(n+m)$ space and answer queries in time $O(\log(n+m))$. It suffices to just explain how the reduction is performed and justify its correctness, query time, and space requirements. (You do not need to explain how to compute the segments nor how to construct the point-location data structure.)

Hint: Begin by extracting the equation for the line ℓ carrying the query ray. Then show that by applying duality, this problem can be reduced to a point-location in $\mathbb{R}^2$ where the segments are determined by the segment vertices A and B. It may be easier to ignore the top and bottom sides of the rectangle, and imagine that the segments extend infinitely up or down. Once the result is know, consider the affect of the rectangle sides. Another useful simplification is to describe two separate data structures, one for A and the other for B, and then combine the results.

Problem 3. 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. 3(a) we show a triangulation of a set of $v = 16$ vertices with $h = 10$ vertices on the convex hull. In Fig. 3(b) we show a quadrilateral cell complex. Using Euler's formula, answer each of the following questions.



(a) (b)

Figure 3: Problem 2: Applications of Euler's formula.

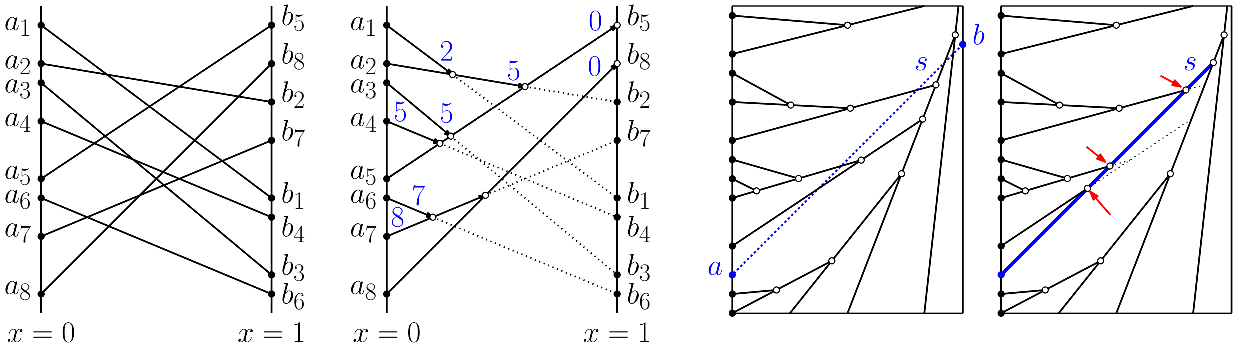
(a) Given a triangulation with v vertices, where h of these lie on the convex hull, use Euler's formula to derive the formula for the number of triangles t and the number of edges e in the triangulation as functions of v and h. (**Hint:** Using the fact that each triangle is incident to three edges, derive a formula that expresses t as an exact function of e and h.)

(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 4. (Optional–Ungraded) 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. 4(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. 4(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. 4(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) (b) (c) (d)

Figure 4: Terminating segments

(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. 4(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.)

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: Suppose you have an infinite chessboard on the x, y-coordinate plane. You may place a finite number of pieces (as many as you like), one per cell, anywhere on the board, but must all to the left of the y-axis (see Fig. 5). You may then make a series of moves, where each consists of one piece jumping over an adjacent piece into an empty cell, vertically or horizontally (but not diagonally).

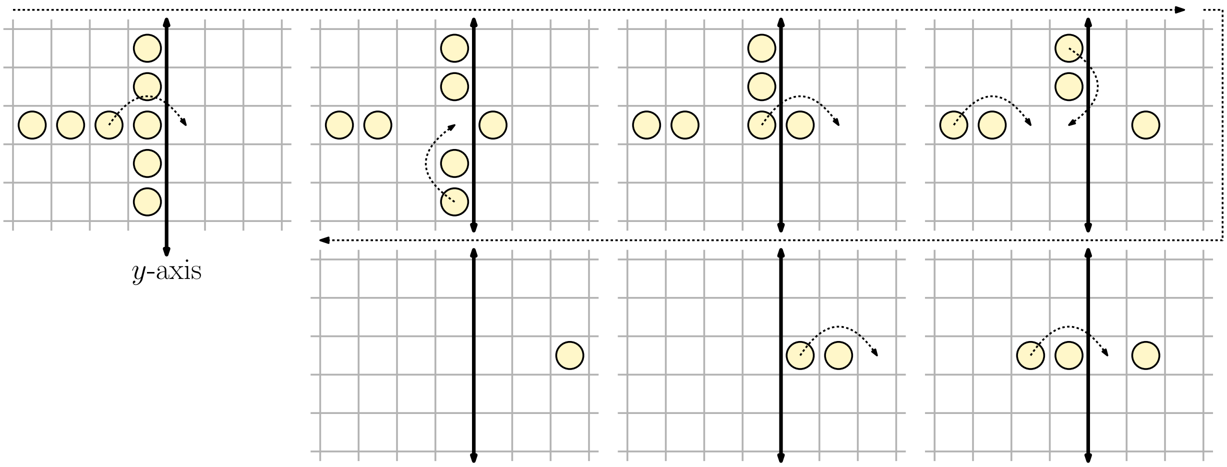


Figure 5: Chessboard jumping

Our objective is to move a piece as far as possible to the right of the y-axis. In Fig. 5, we show an initial set of 8 pieces, where it is possible to move a piece 3 columns from the y-axis. Presumably, as you add more pieces, you would hope to be able to get a piece farther and farther from the y-axis, but surprisingly, this not so!

Prove that, no matter what the initial number or configuration of pieces is and no matter what sequence of moves is made, there is an integer $c > 0$, such that it is impossible to move any piece into a cell beyond the c-th column to the right of the y-axis.

Spoiler alert: If you want to puzzle this out on your own, feel free to ignore this hint. Otherwise, please flip to the next page.

The proof that I know of is based on a *potential argument.* Let us assume that there is a designated *target cell* t on the right side of the y-axis, which we want to reach. Let $t_x \geq 1$ denote its column number. Given any set S of pieces on the board, we will define a real-valued potential function $\Phi(S)$, which will have the following properties:

Distance based: The function is based on a constant φ, where $0 < \varphi < 1$. Cell s of the chessboard is a assigned a *value* of the form φ^{ℓ}, where ℓ is the Manhattan distance (sum of vertical and horizontal components) from s to t. Observe that t itself has a value of $\varphi^0 = 1$.

Accumulative: For any set S of placement of board pieces, define $\Phi(S)$ to be the sum of the values of the cells where these pieces reside.

Nonincreasing: Whenever a move is made, mapping some set S to a set S', we have $\Phi(S') \leq \Phi(S)$.

The proof involves two principal steps:

(a) Derive a value of φ to satisfy the above requirements, particularly that the potential is nonincreasing. (**Hint:** There are three cases, depending on whether the jump causes the distance to the target to increase, decrease, or remain unchanged. The critical case is when the distance is decreasing. While there are many choices for φ that work, the smallest one yields the best bound on c. It turns out to be related to the Golden ratio.)

(b) Prove (through the use of the geometric series) that if t_x is greater than some constant, the total potential of any (even an infinite) set S that lies entirely to the left of the y-axis is strictly smaller than 1.