Final exam CS 498sh3: Computational Geometry, Spring 2023

May 10, 2023, Wednesday, 8am, 1304 SC

Instructions:
(A) There are many (five, but who counts?) questions in this exam. You should answer all of them.
(B) This is a closed book exam. You are allowed a hand written cheat sheet, and that is it.
(C) The exam lasts 180 minutes ±30 seconds.
(D) Please print your name, netid, email, etc in the boxes provided below.

Name	
Netid	
Email	

1 (20 PTS.) An easier variant of practice exam question 1.

`≪Long tedious explanation motivating the problem, involving tigers, aliens, ice cream, and the latest Bond movie villain.≫`

The input is a set S of n axis parallel rectangles (in the plane) all with their bottom edge lying on the x-axis. Describe an algorithm, as fast as possible (but not faster than that), that computes the skyline formed by these rectangles.

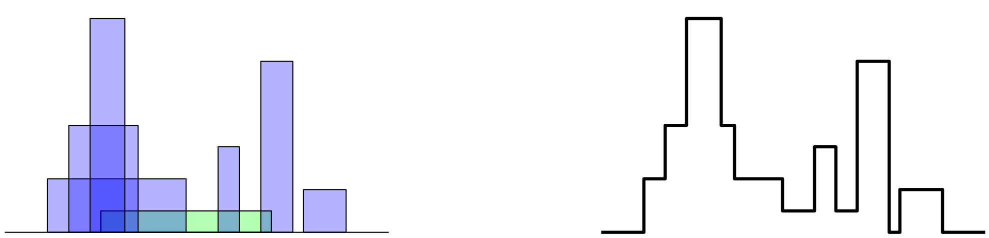

Figure 0.1: Left: Input. Right: Output.

Reminder: Write short concise answer.

2 (20 PTS.) Question 1.8 from the book.

Let P be a set of n points in the plane. Present a divide-and-conquer algorithm for computing the convex-hull of P. For credit your algorithm running time has to be optimal.

2.A. (10 PTS.) Let P_1 and P_2 be two disjoint convex polygons with n vertices in total. Describe a *deterministic* $O(n)$ time algorithm that computes the convex hull of $V(P_1) \cup V(P_2)$. For simplicity, you can assume that all the points of P_1 are to the left (on the x-axis) to the points of P_2. (There is a short an elegant algorithm here, BTW.)

2.B. (10 PTS.) Use the algorithm from the previous part to describe an $O(n \log n)$ time divide-and-conquer algorithm to compute the convex hull of a set of n points in the plane. Prove the running time of your algorithm.

3 (20 PTS.) Problem 3.13 from the book.

The ***stabbing number*** of a triangulated simple polygon P is the maximum number of diagonals intersected by any line. Give an optimal algorithm that computes a triangulation of a given convex polygon (that has n vertices) that has stabbing number $O(\log n)$. Prove the bound. What is the running time of your algorithm?

4 (20 PTS.) Problem 4.16 from the book.

On n parallel railway tracks, n trains (choo choo!) are going with constant speeds $v_1, v_2, \ldots, v_n > 0$, respectively. At time $t = 0$, the trains are at positions $k_1, k_2, \ldots, k_n$, respectively. Give an $O(n \log n)$ time algorithm that detects all trains that at some moment in time are leading (i.e., if the rails are parallel to the x-axis, a train is *leading* if its x value is maximal, among all trains, at this point in time). You can assume the trains are yellow, if it helps.

5 (20 PTS.) An easier variant of problem 15.6 from the book.

Let P be a given x-monotone polygon with n vertices in the plane. That is, every vertical line intersects P in a segment (or not at all). Let s and t be the leftmost and rightmost vertices of P. Describe an algorithm, as fast as possible, that computes the shortest path in P between s and t. For any credit, your algorithm should run in subquadratic time. Full credit would be given only to an optimal algorithm. What is the running time of your algorithm?

(BTW, using this algorithm, it is not hard to solve the general case, where the polygon is not necessarily monotone.)

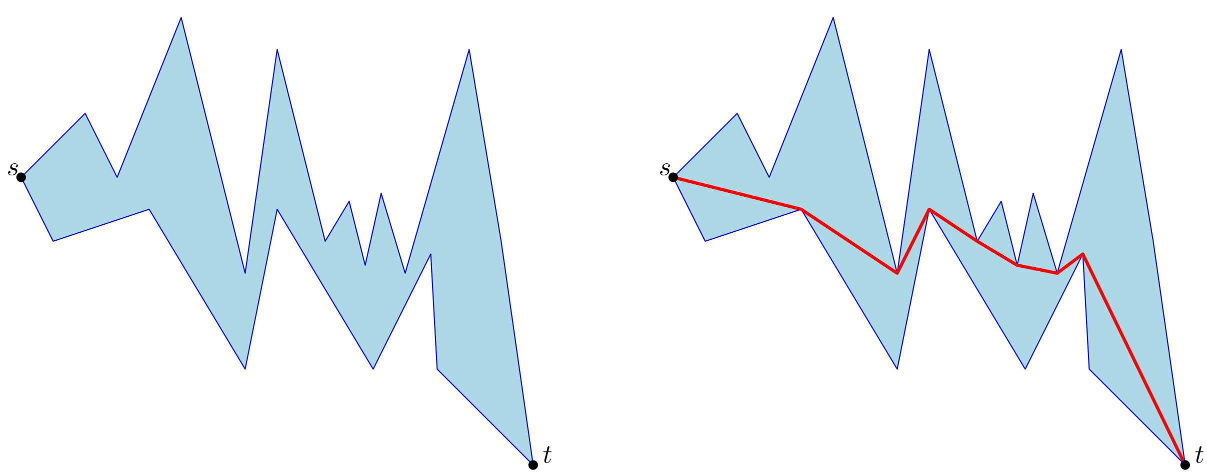

Figure 0.2: Left: Input. Right: Output.