COMPUTATIONAL GEOMETRY - FINAL TAKE-HOME EXAM

Prof. Micha Sharir
Department of Computer Science
Raymond and Beverly Sackler Faculty of Exact Sciences
January 20–23, 2005

Answer four of the following problems. All problems have equal weight (25 percent). The exam should be returned to Ruti Friedberg (mazkirat toar sheni, Kaplun, 3rd floor) by 12:00 noon Sunday, January 23. Questions concerning the exam can be addressed by email to both `michas@post.tau.ac.il` and `estere@post.tau.ac.il`. Watch also the web page, especially for comments or clarifications (if needed). Good luck!!

Problem 1

Let S be a set of n line segments in the plane in general position. Preprocess S into a data structure so that, given a query line ℓ, one can determine efficiently whether ℓ *separates* S. This means that ℓ does not intersect any segment and there is at least one segment of S on each side of ℓ. The goal is to achieve $O(\log n)$ query time. What are the storage and preprocessing costs of your algorithm? (**Hint:** Use duality: express the relations between ℓ and a segment of S in terms of the dual point of ℓ.)

Problem 2

Let P be a set of n points in three dimensions, and let ℓ be a line. Show how to find in linear time the two points of intersection of ℓ with the (boundary of the) convex hull of P, if these points exist (or to determine that ℓ does not intersect the hull). Show also how to extend the problem and the solution to any fixed dimension d. (**Hint:** Restate the problem as a linear programming problem.)

Problem 3

Let P be a convex polygon with n edges in the plane, which contains the origin in its interior. Consider each edge of P as a site, and let $\mathrm{Vor}(P)$ denote the Voronoi diagram of these n line segments, *restricted to the interior of* P.

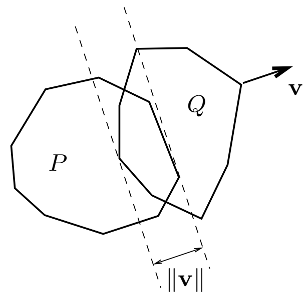
(a) What is the shape of each Voronoi edge and cell of the diagram? What is the combinatorial complexity of the diagram?

(b) Show that, within P, the above Voronoi diagram is identical to the Voronoi diagram of the n lines that contain the edges of P. (That is, if the edge of P nearest to a point w inside P is e, among all edges of P, then the line supporting e is the nearest line to w, among all lines that support the edges of P.)

(c) Using (b), write in explicit form the n functions that measure the distances from any point (x, y) within P to these n lines, follow the approach of interpreting the Voronoi diagram as equivalent to the pointwise minimum of these n functions, and derive from this an $O(n \log n)$ algorithm for constructing $\mathrm{Vor}(P)$. (Alternatively, derive any other equally efficient algorithm for constructing the diagram.)

Problem 4

Let P and Q be two *intersecting* convex polytopes in $\mathbb{R}^3$, so that P has m facets and Q has n facets. The *penetration depth* of P and Q is the length of the shortest vector $\mathbf{v}$ such that the interiors of P and of $Q + \mathbf{v}$ are disjoint. (That is, we want to translate Q (without rotation) in a straight direction, so as to separate it from P, and the penetration depth is the shortest length of such a separating motion.)

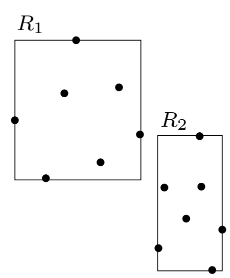


(a) Show that, for any direction $\mathbf{u}$, the length of a shortest separating motion in direction $\mathbf{u}$ is the distance between the two planes π_1, π_2 that are orthogonal to $\mathbf{u}$ and support P and Q respectively, such that the normal direction $\mathbf{u}$ of π_1 (resp., π_2) points *away* from P (resp., *into* Q).

(b) Use the normal diagrams of P and Q to find, in time $O(mn \log(m + n))$, the shortest separating vector $\mathbf{v}$ and its length (the penetration depth). (**Hint:** Overlay the diagrams and show that the best direction is a *vertex* of the overlay.)

Problem 5

Let P be a set of n points in the plane. Find, in $O(n \log n)$ time, two disjoint axis-parallel rectangles R_1, R_2, such that R_1 lies fully to the left of R_2, such that $P \subset R_1 \cup R_2$, and the sum of the areas of R_1 and R_2 is as small as possible. (**Hint:** Use sweeping.)



Problem 6

Let T be a given set of n horizontal triangles in three dimensions, each lying at different z-coordinate, all of which are positive. Give an algorithm for computing the *visibility map* of T. This is a subdivision of the xy-plane into maximal connected regions, so that for each region R there is a unique triangle $t \in T$, such that for all $(x, y) \in R$ the z-vertical line through (x, y) intersects t below any other triangle (the point (x, y) 'sees' t when looking upwards). What is the worst-case running time of the algorithm. (The output, a planar map, should be in DCEL format, with each face labeled by the triangle that it sees, or by a flag that indicates that it sees no triangle.)