Assignment 3 - Computational Geometry (0368-4211)

Due: December 29, 2008

Problem 1

Given n non-intersecting and non-vertical line segments $e_1, \dots, e_n$ in the plane. Preprocess them so that, given any vertical line segment b, one can count quickly how many segments intersect b.

Problem 2

Let P be a set of n points in the plane in general position. Preprocess P into a data structure of linear size which can answer the following kind of queries in $O(\log n)$ time: In each query we are given a point q in the plane and a real number a, and we wish to determine whether q lies in the convex hull of the subset P_a of those points of P that lie to the left of $x = a$. (**Hints:** (a) Reduce the problem to the simpler problem where we need to determine whether q lies below the upper convex hull of P_a. (b) Think of the Graham walk, and use persistence to solve the latter problem.)

Problem 3

Given a set $T = \{\Delta_1, \Delta_2, \dots, \Delta_n\}$ of n horizontal triangles in three dimensions, so that Δ_i lies in the plane $z = i$, for each i. Preprocess T into a data structure that supports efficiently queries of the form: Given a point q in 3-space, find the triangle of T lying vertically directly below q (this is the first triangle that we see when we stand at q and look down, in the negative z-direction). The goal is to achieve query time of $O(\log n)$. How much storage and preprocessing does your solution require? Express the answer in terms of n and the number k of intersections between the edges of the projections of the triangles on the xy-plane. (**Hint:** Use planar point location in the xy-plane, plus persistence of the structure over each face of the planar map.)

Problem 4

Let P be a convex polytope with n facets in three dimensions. Preprocess P into a linear-size data structure so that, given any query plane H, we can determine, in

$O(\log n)$ time, whether H intersects P. (**Hint:** Use the normal diagram of P, as defined in a previous exercise.)

Problem 5

(a) Let L be a set of n lines in the plane. Preprocess L into a data structure of size $O(n^2)$, so that, given any query point q, one can compute, in $O(\log n)$ time, the number of lines of L below q.

(b) Suppose we know in advance the number m of queries. Modify the solution in (a) so that the total running time for answering the queries (including preprocessing cost) is roughly $O(n\sqrt{m} + m)$ (up to polylogarithmic factors).

(c) Modify the solution in (b) so that it also works when we do not know the number m in advance. In other words, for any m, after m queries have arrived, the total time spent so far should be roughly $O(n\sqrt{m} + m)$ (up to polylogarithmic factors).