Assignment 3 - Computational Geometry (0368-4211)

Due: May 29, 2015

Problem 1

Let S be a set of n possibly intersecting line segments in the plane. Construct a data structure that, given a query point q, finds the segment of S lying vertically directly above q. Use the persistence search tree technique, so as to achieve $O(\log n)$ query time. Bound the storage and preprocessing time in terms of n and of $k =$ the number of intersections among the segments of S.

Problem 2

Let $S = \{e_1, \ldots, e_n\}$ be a set of n pairwise disjoint horizontal segments in the plane (in general position). Construct the vertical decomposition of the plane into trapezoids (rectangles in this case) formed by S, as defined in class, using randomized incremental construction: Insert the segments one by one in a random order, and maintain the decomposition after each insertion. To facilitate the process, each current trapezoid maintains a "conflict list" of all segments not yet inserted that intersect it, and, conversely, each segment not yet inserted maintains a list of the current trapezoids that it intersects.

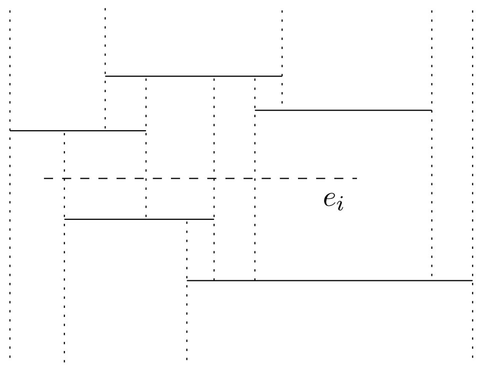

Figure 1: The insertion of e_i into the present trapezoidal decomposition.

(a) Describe the procedure for inserting a new segment e_i into the current decomposition (see Figure 1), including how to construct the new rectangles, and how to construct their conflict lists from those of the destroyed rectangles. Argue that the first step can be done in time proportional to the number of current rectangles that e_i intersects, and that the second step can be done in time proportional to the size of their conflict lists.

(b) Using a suitable variant of the analysis that we applied to the randomized incremental construction of convex hulls, show that the expected number of rectangles that the algorithm generates is $O(n)$, and that the expected running time of the whole procedure is $O(n \log n)$. (**Hint:** You need to consider all possible rectangles that can ever arise, define the "weight" of each such rectangle, and apply the (suitable variants of the) probabilistic calculations that we did in class.)

(c) Using a suitable variant of (b), show the following: Let q be a fixed point in the plane (not lying on any segment). Show that the expected number of trapezoids that the algorithm generates that contain q is $O(\log n)$. (**Hint:** The only real change is in the application of the Clarkson-Shor technique.)

(d) **Bonus**, if you still have energy: Try to combine (a)–(c) to turn the construction into a data structure for efficient point location in S. Explain how the search with a query point q is implemented.

Problem 3

Use persistent search trees to solve the following problem. Let $e_1, e_2, \ldots, e_n$ be n pairwise disjoint line segments in the plane, all lying above the x-axis. Preprocess them into a data structure of size $O(n^2)$, such that, given any query ray ρ that starts at any point on the x-axis and goes into the upper halfplane $y > 0$ in any direction θ (that is, the query is given by the pair (x, θ), where x is the x-coordinate of the starting point and θ is the direction of the ray), we can determine, in $O(\log n)$ time, the first segment hit by the ray, or determine that it does not hit any segment.

Hint: First solve the problem when one of the parameters is fixed (say, θ is fixed). Then consider what happens to the structure as θ changes, and use persistence to record the changes.