Homework #3: Point location, polygon triangulation [60 points]
Due Date: in class, on Wednesday, 1 June 2005 – ** no late days for this homework **

- **The Common Theory Problems**

Problem 1. [10 points]

Consider a subdivision of the plane consisting entirely of rectangles aligned with the x- and y-axes. Such a subdivision is clearly monotone for any direction θ, $0 < \theta < \pi/2$ (in other words, a line in the direction θ always cuts each region along at most a single segment). By what we proved in class, it follows that, for each fixed θ, the corresponding "above" relation is acyclic. Prove that there is actually a linear ordering of the regions that is consistent with the "above" relations *for all* θ*,* $0 < \theta < \pi/2$*, simultaneously.*

Problem 2. [10 points]

We discussed briefly in class the point location method due to David Kirpatrick which constructs a hierarchy of coarser and coarser triangular subdivisions over the original triangulation [*SIAM J. Comp., 12 (1983), 28–35*]. Adapt his method to perform point location on subdivisions consisting entirely of rectangles aligned with the axes. Your method should build a hierarchy of subdivisions that are all of the same type as the original: edges must be vertical or horizontal and regions rectangular. Your asymptotic preprocessing, space, and query bounds should be the same as Kirkpatrick's. (*Hint:* You may want to consider removing elements other that vertices in the coarsening process.)

- **The Additional Theory Problems**

Problem 3. [10 points]

Read Section B1 of the "Ruler, Compass, and Computer" paper (included in the course reader). Show how the interval stabbing structure presented there can be adapted to report all the intervals containing the query point, not just count them. The time for the reporting operation should be $O(\log n + k)$, where k is the number of intervals reported. Then show that counting and reporting can also be done (within the same time bounds) when the query object is not a point but another interval, and we are interested in the original intervals intersecting the query interval. The more ambitious of you can now try to use this method plus a sweep line idea to give an $O(n \log n + k)$ algorithm for reporting all intersecting pairs among n axis-aligned rectangles in the plane (again k is the output size) [5 extra points]. This is a very useful operation in design rule checking for VLSI circuits.

Problem 4. [10 points]

Let S be the set of vertices of a simple polygon P in the plane and call a diagonal AB of P *extreme* if both A and B are vertices of the convex hull of S. Show that, unless P is convex, P can be triangulated without using any extreme diagonals.

Problem 5. [10 points]

Let P be a simple polygon on n sides. Show how to compute the vertical trapezoidalization of P in linear time, starting from an arbitrary triangulation of P (we covered the other direction in class).

Problem 6. [10 points]

Let P be a simple polygon on n sides. Consider a particular side e of P as a luminous neon tube, casting light towards the interior of P. The illuminated area of P is another polygon V, called the *weak-visibility* polygon of P from e. Every point of V has the property that "it can see" some point of e. Show how to compute V in linear time, starting from an arbitrary triangulation of P.

- **The Programming Problem**

Problem 7. [40 points]

The third programming problem will be designed individually to accommodate the diverse interests of the students in the applied track. For example, many extensions are possible along the lines of the kinetic Delaunay triangulation of homework 2. Other 2-D geometric structures on points that are interesting to maintain under motion include the closest or furthest pair, the minimum spanning tree, etc. When modeling moving segments, problems of interest include maintenance of the vertical decomposition, of a binary space partition tree, or of the visibility complex. It is also perfectly OK to submit a project that is related to, or part of, another piece of work that you are doing and which involves geometric algorithms. In that case you should take care to explain the part of the work that involves the types of techniques addressed in this class.

A one or two page proposal about the last programming assignment is due on Monday, 16 May 2005.