Assignment 4
CSCI-4963/6962: Geometric Algorithms
Due: Thursday, March 2, 2000

Assignments are due at the beginning of class on March 2, and are to be done individually. Assignments will be graded on the basis of correctness, clarity, and legibility. Late assignments incur a 10% penalty. Assignments handed in more than a week late will receive no credit.

This assignment focuses on polygon triangulation and point location.

1. Prove or disprove: The dual graph of the triangulation of a monotone polygon is always a chain, that is, any node in this graph has degree at most two.

2. The *pockets* of a simple polygon are the areas outside the polygon, but inside its convex hull. Let P_1 be a simple polygon with n_1 vertices, and assume that a triangulation of P_1 as well as of its pockets is given. Let P_2 be a convex polygon with n_2 vertices. Show that the intersection $P_1 \cap P_2$ can be computed in $O(n_1 + n_2)$ time.

3. Given a simple polygon P with n vertices and a point p inside it, show how to compute the region inside P that is visible from p.

4. Consider the *single shot* problem, where the subdivision and query point are given at the same time. Given a simple polygon P with n vertices and a query point q, consider the horizontal ray $\rho = \{(q_x + \lambda, q_y) : \lambda > 0\}$ starting at q. If the number of edges of P intersected by this ray is odd, then $q \in P$, otherwise $q \notin P$.

   Given a convex polygon P as an array of its n vertices in sorted order along the boundary, show that it can be tested in time $O(\log n)$ whether a query point q lies inside P.

5. Prove that the number of inner nodes of the search structure $\mathcal{D}$ of algorithm TrapezoidalMap increases by $k_i - 1$ in iteration i, where k_i is the number of new trapezoids in $\mathcal{T}(S_i)$ (and hence the number of new leaves of $\mathcal{D}$).

6. The *ray shooting problem* occurs in computer graphics, where the visible object at each pixel is to be determined. A 2-dimensional version is: Store a set S of n non-crossing line segments such that one can quickly answer queries of the type: "Given a query ray ρ, find the first segment in S intersected by ρ. Consider vertical ray shooting, where the query ray must be a vertical ray pointing upwards. Give a data structure for the vertical ray shooting problem for a set S of n non-crossing line segments in general position. Bound the query time and storage requirements of your data structure. What is the preprocessing time?