CS 6463-004 Computational Geometry, Fall 06

9/21/06

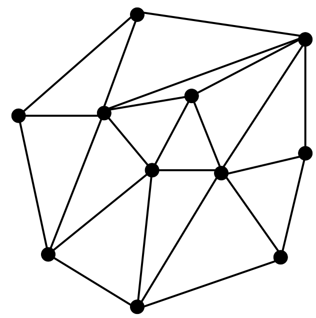
2. Homework

Due **10/5/06** before class

Always justify the runtime and the correctness of your algorithms, and try to make algorithms as efficient as possible.

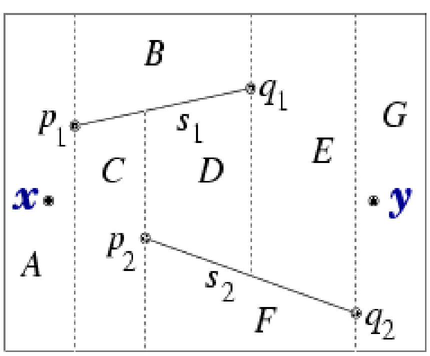
1. (3.10) Triangulating a Point Set – 10 points

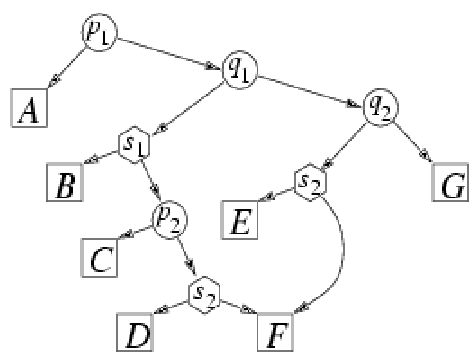
   A triangulation of a set of points P in the plane is a decomposition of $CH(P)$ into triangles whose vertices are points of P and which contain no point of P in the interior of a triangle or an edge.
   Explain how to adapt the discussed triangulation algorithm to efficiently triangulate a set of n points.

2. Monotone Polygon – 5 points
   Give an example of a simple polygon which is x-monotone and y-monotone, but which is neither monotone with respect to the line $y = x$ nor monotone with respect to the line $y = -x$.

3. Trapezoidal Map Example – 5 points
   Consider the following instance of the trapezoidal map point location data structure. The left side shows the map, and the right side shows the corresponding search structure. Describe how the search structure is modified if the next segment to be added is $\overline{xy}$.

4. Point Location Without Preprocessing – 20 points
   Consider the following *single shot* problems, where the subdivision and the query point are given at the same time, and no preprocessing is allowed to speed up the query time.

   a) (6.4)(10pts) Show that, given a planar subdivision $\mathcal{S}$ (given in a doubly-connected edge list) with n vertices and edges and a query point q, the face of $\mathcal{S}$ containing q can be computed in time $O(n)$.

   b) (6.5)(10pts) Given a convex polygon $\mathcal{P}$ as an array of its n vertices in sorted order along the boundary. Show that, given a query point q, it can be tested in time $O(\log n)$ whether q lies inside $\mathcal{P}$.