6.854J / 18.415J Advanced Algorithms
Fall 2008

18.415/6.854 Advanced Algorithms

Problem Set 6

1. In lecture, we saw a randomized incremental algorithm to find the convex hull of n points in $\mathbb{R}^d$. The algorithm first selects a random ordering, say $p_1, \cdots, p_n$, of the n points, and then incrementally computes $P_i = conv(\{p_1, \cdots, p_i\})$. When going from P_{i-1} to P_i, some of the facets of P_{i-1} disappears (the ones that were visible from x_i) and a number of new facets are created. Let N_i be the number of facets created at step i. The expected running time of the algorithm can be shown to be $O(n^2 + \sum_{i=d+2}^{n} E[N_i])$. Prove that

$$\sum_{i=d+2}^{n} E[N_i] = O\left(n^{\lfloor d/2 \rfloor}\right).$$

(You can use the fact that the convex hull of k points in $\mathbb{R}^d$ has $O(k^{\lfloor d/2 \rfloor})$ facets.)

2. We haven't discussed this in lecture, but given a Voronoi diagram, one can construct in $O(n)$ time a data structure for point location with query time $O(\log n)$: a query consists of a point $q \in \mathbb{R}^2$ and the output sahould be the Voronoi cell that contains this point (this is known as point location). Although we haven't discussed this data structure in lecture, let's assume we have such a data structure.

   Now, suppose we are given two sets of n points in $\mathbb{R}^2$: $P = \{p_1, \cdots, p_n\}$ and $Q = \{q_1, q_2, \cdots, q_n\}$. Define the Hausdorff distance between A and B to be:

$$d(A, B) = \max\left(\max_i \min_j ||p_i - q_j||, \max_j \min_i ||p_i - q_j||\right)$$

   where $||x||$ is the standard Euclidean norm. Describe an algorithm to compute $d(A, B)$ in $O(n \log n)$ time.

3. Suppose you are implementing a video game in which the player can walk around a planar environment made up of walls, and at any time the screen displays only the walls that are (partially) visible by the player. More precisely, the player is modeled as a single point; the walls are modeled as noncrossing line segments; two points are *visible* if the line segment connecting them does not intersect any walls except at its endpoints; and a wall is *visible* from a point if at least one point on the wall is visible from the point. Give an $O(n \lg n)$-time algorithm

to compute the set of walls visible from the player. **Hint**: Use a line-sweep algorithm, but instead of sweeping a horizontal line, sweep a half-line around a point.

4. Given a set $P = \{p_1, p_2, \cdots, p_n\}$ of points in $\mathbb{R}^2$, the Delaunay graph $D(P) = (P, E)$ is a graph whose vertex set is P and which has an edge $(p_i, p_j) \in E$ if and only if part of the bisector between p_i and p_j is an edge (line segment) of the Voronoi diagram $Vor(P)$.

   (a) Show that (p_i, p_j) is an edge of $D(P)$ if and only if there exists an empty circle (i.e. no points of P in its interior) with only p_i and p_j on its boundary. (This is easy; you can rely on properties derived for the Voronoi diagram in lecture.)

   (b) Given P, the *Euclidean minimum spanning tree problem* is the minimum spanning tree problem in the complete graph whose vertex set is P and with the length of the edge (p_i, p_j) equal to its Euclidean length. Prove that if T is a Euclidean minimum spanning tree then $T \subseteq E$ where E is the edge set of the Delaunay graph $D(P)$.

   (Hint: What is a property of any edge of a minimum spanning tree?)

   (c) How efficiently can you find the Euclidean minimum spanning tree of a set P of n points?

   (d) The definition of the Delaunay graph $D(P)$ shows that it is dual to the Voronoi diagram $Vor(P)$ in the (planar graph) sense that two points p_i and p_j are connected in $D(P)$ iff their corresponding cells are adjacent (i.e. share an edge) in $Vor(P)$. This implies that $D(P)$ is a planar graph since its edges can be drawn by non-intersecting curves. Show that we do not even need to move the (positions of the) points in P to get a straight-line drawing of $D(P)$, i.e. that if (p_i, p_j) and (p_k, p_l) are edges of $D(P)$ then the corresponding line segments do not intersect (except at an endpoint if $\{i, j\} \cap \{k, l\} \neq \emptyset$).