CSL 852 Computational Geometry

Midterm , Sem I 2010-11, Max 50, Time 2 hrs

Note (i) Write your answers neatly and precisely. You won't get a second chance to explain what you have written.
(ii) Every algorithm must be accompanied by proof of correctness and a formal analysis of running time and space bound. Feel free to quote any result from the lectures without proof - for any anything new, you must prove it first.

1. Given a set S of n line segments (mutually non-intersecting), construct a data structure that supports a query of the following kind -
   For any arbitrary point p, find the segment that lies immediately above p (called vertical ray shooting query)
   (i) Design a data structure based on segment trees that answers such queries quickly.
   (ii) **Bonus** Can you present a scheme that answers the queries in $O(\log n)$ time ? **(15 marks)**
   Construct a segment tree and within each node v build a data structure that supports vertical ray shooting query for the segments stored in v, sy S_v. Since the segments in S_v are totally ordered (within the interval spanned by v), we can do binary search using a *above-below* primitive in $O(\log n)$ steps. So vertical ray shooting is done by first using a binary search in the x direction that identifies all those nodes V such that $\cup_{v \in V} S_v$ are exactly those segments that intersect the vertical line through the query point q. The set V is simply the search path of q in the segment tree. Then we do binary searches in all nodes of V and report the closest segment in the upward direction. Overall it takes $O(\log^2 n)$ for query and $O(n \log n)$ space.

   To improve the bounds, you can use line sweep to build the trapezoidal map of the set of line segments using vertical visibility information during the line sweep process. Then build the Dobkin-Kirkpartick planar point location data structure that can answer a query in $O(\log n)$ time. If we know the trapezoid, we also know the vertical visbility segment. The preprocessing takes $O(n \log n)$ time and the space is $O(n)$ - the total size of all trapezoids.

2. Let $\mathcal{M}(S)$ denote the set of maximal points of a planar point set S. Denote $L_0 = \mathcal{M}(S)$ and $S_0 = S$ and let $S_i = S_{i-1} - L_{i-1}$ and $L_i = \mathcal{M}(S_i)$ for $i \geq 1$.
   You can think about L_i's as the *maximal layers* that are successively obtained by stripping away the previous layers. Design an $O(npolylog(n))$ algorithm for computing all the maximal layers. **(10 marks)**
   Do a line sweep in the decreasing order of x (i.e. sort the points on their x coordinate value) let this sorted set be $p'1, p'_2 \dots p'_n$. Initialize $L_0 = p'_n$ and as we sweep left, assume that we have inductively computed the layers correctly till p'_{i+1}. When we consider p'_i then suppose the layers are $L_0, L_2 \dots L_j$ and let $Y_0, Y_1, \dots Y_j$ denote the highest y coordinates of the points in the respective layers.
   Claim: p'_i belongs to L_k iff $Y_{k+1} > y'_i > Y_k$ if such a k exists or start a new layer $j+1$ if $y'_i < Y_j$
   Using a dynamic dictionary, this can be found in $O(\log n)$ steps and therefore the entire algorithm takes $O(n \log n)$ time.

3. It is known that if the n input points are distributed uniformly at random in a unit disk then the expected number of points on the hull is less than $\sqrt{n}$. Based on this property, design an $O(n)$ algorithm for convex hull. Note that the running time will be expected $O(n)$ over the distribution of the input points and not for the *worst case* input.
   Hint: A subset of uniformly distributed points is also uniformly distributed in a disk and has the same property. **(10 marks)**
   You may think about the each point as being generated independently at random from the distribution, so the first $n/2$ points have identical distribution to the last $n/2$. Therefore construct the convex hull of the first $n/2$ points $CH1$ recursively and the remaining $n/2$ points $CH2$ recursively, and merge them. Note that the expected number of points in $CH1$ and $CH2$ are $O(\sqrt{n}$. So we can

construct the hull of the union of $CH1$ and $CH2$ in $O(\sqrt{n})$ steps - even though they may not be linearly separable.
This leads to $T(n) = 2T(n/2) + O(\sqrt{n})$ where $T()$ is the expected running time. The solution for this is $T(n) = O(n)$ and it can be verified by induction.

4. Let $S = \{p_1, \ldots, p_n\}$ be a set of n points in the plane so that no three of them lie on a line. The farthest-point Voronoi diagram of S is planar decomposition of the plane into maximal cells so that the same point of S is the farthest neighbor of all points within each cell. That is, it is the decomposition induced by the cells

$$\text{Vor}_f(p_i) = \{x \in R^2 \mid \|p_i x\| \geq \|p_j x\| \forall j\}.$$

   (i) Show that $\text{Vor}_f(p_i)$ is convex.

   (ii) Show that $\text{Vor}_f(p_i)$ is nonempty if and only if p_i is a vertex of the convex hull of S.

   (iii) Show that if $\text{Vor}_f(p_i)$ is nonempty then it is unbounded.

   (15 marks)

(i) Fix a point $p_i \in S$, for any other point $p_j \in S$, let h_j^- denote the half-space defined by the perpendicular bisector of p_i, p_j, not containing p_i. The Voronoi cell of p_i, is the intersection of half-spaces h_j^-, i.e., $\text{Vor}_f(p_i) = \bigcap_{j \neq i} h_j^-$, and therefore it is convex. **(3 marks)**

(ii) Part A: If p_i is a vertex of the convex hull of S, then $\text{Vor}_f(p_i)$ is non-empty.
Since no three points of S are collinear, we can choose a tangent τ to the convex hull at p_i that is not parallel to any side of the boundary of the convex hull. Let ρ be the perpendicular to τ at p_i. As we move along ρ such that the distance to p_i increases, we can find a point q on ρ such that the disk of radius $||qp_i||$ contains every point of the convex hull, other than p_i, in its interior.
So $||qp_i|| \geq ||qp_j||$, for all $i \neq j$, so $q \in \text{Vor}_f(p_i)$.
Part B: If p_i is not a vertex of convex hull of S, then $\text{Vor}_f(p_i)$ is empty.
Suppose for some p_i that is not a vertex of convex hull of S, there is an $x \in \text{Vor}_f(p_i)$. Let y be the point of intersection of the ray $\overrightarrow{xp_i}$ with the boundary of the convex hull of S such that y does not lie on the segment $\overline{xp_i}$. Using triangular inequality, it can be shown that one of the two vertices v of the convex hull that is adjacent to y is farther than p_i from x. Contradiction.

(6 marks)

(iii) Following the same argument as in (ii), Part A, for any point p_i on the convex hull we can find points q on the perpendicular ρ such that $q \in \text{Vor}_f(p_i)$, as we move away from p_i along ρ. So the Voronoi region is unbounded. **(6 marks)**