CS 6463 Computational Geometry, Fall 10

10/20/10

4. Homework

Due **11/10/10** before class

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

1. Convex Hull of Intersections (15 points)
   Let $\mathcal{L}$ be a set of n lines in the plane, no two of which are parallel. Let S be the set of all $O(n^2)$ intersection points between any two lines in $\mathcal{L}$.

   (a) Give an $O(n \log n)$ time algorithm to compute an axis-parallel rectangle that contains S.

   (b) Give an $O(n \log n)$ time algorithm that computes $CH(S)$.

   Hint: Your algorithms cannot compute all points in S explicitly. Sort all lines by slope, and prove that it is enough to consider only a certain subset of intersection points.

2. Star-Shaped Polygon (10 points)
   A simple polygon P is called *star-shaped*, if it contains a point c such that for any point $p \in P$ the line segment $\overline{cp}$ is contained in P.

   (a) Give an example of a star-shaped polygon that is not convex. What do you think is the reason for the name *star-shaped*?

   (b) Develop an algorithm with expected linear runtime to decide whether a simple polygon is star-shaped.
   (Hint: Half-plane intersection.)

3. Linear Separator (10 points)
   Given m red points $R = \{r_1, \ldots, r_m\}$ and n blue points $B = \{b_1, \ldots, b_n\}$ in the plane. The **linear separator problem** is to decide whether there exists a line l such that all points of R are on one side of l and all points of B are on the other side. (You may assume appropriate general position, and may disregard points that lie exactly on the line.)

   Use point-line duality to develop an algorithm for this problem which runs in expected linear time. *(Hint: Linear Programming.)*

4. Worst-case LP (5 points)
   Suppose you are given a linear program consisting of n distinct half-planes and a linear objective function. Assume the linear program is bounded.

   Is there always a way to order the half-planes such that the algorithm requires at least $\Omega(n^2)$ time? If so, explain how to construct such an ordering (given *any* fixed set of n half-planes). If not, give an example for arbitrary n in which all orderings lead to an $o(n^2)$ runtime.