15-451 Algorithms, Spring 2012

Homework 4 (100 pts) Due: Mar 07

Question	Points	Score
1	25	
2	25	
3	20	
4	30	
Total:	100	

1. **Minkowski set**

Given two convex polygons, P and Q with n vertices. The Minkowski sum $M = P + Q$ is the set $\{x + y, x \in P \ \&\& \ y \in Q\}$.

(10) (a) How do you compute Minkowski sum M? You may assume that vertices P and Q are ordered.

(5) (b) What is the worst-case runtime complexity of your algorithm in part a)?

(10) (c) Prove that Minkowski sum M is a convex polygon.

2. **Collinearity**

   In computational geometry, we often assume input points are in **General Position**. In the 2D case, this means that points are distinct and no 3 points are collinear. Thus it is important to check for 3 collinear points in a timely fashion. Given n points in 2D, find whether there is 3 collinear points in,

(5) (a) $O(n^3)$ time given $O(1)$ space.

(10) (b) $O(n^2 \log n)$ time given $O(n)$ space.

(10) (c) $O(n^2)$ time given $O(n^2)$ space.

3. **Visibility**

In this problem we are given a collection of n disjoint line segments, $S = S_1, \ldots, S_n$ and a point p. The goal is to determine which segments are visible from p and for each visible segment what subsegments are visible.

(10) (a) Give an $O(n \log n)$ algorithm for this problem.

Hint: try using a radial sweep-line algorithm.

(10) (b) How would you modify your algorithm to work if the segments are not disjoint so that it still has the same asymptotic complexity.

4. **Connecting Dots, aka. Geometric Quick Sort** Given n red and m blue dots on the plane with no three colinear, we want to connect some pairs of dots of the **same color** with straight line segments so all the dots of each color are connected. That is we want a red and a blue spanning tree of line segments where the blue one contains all the blue dots and the red one contains all the red dots. Furthermore, we do not want any of the line segments to intersect. We also want to obtain an algorithm that constructs these two trees in $O((m+n)\log(m+n))$ time.

(5) (a) Give a configuration of points for which this can't be done (hint: 4 points will suffice)

(5) (b) Define a triangle to be nice if its three corners are not of the same color. Show that if we have a nice triangle where all points in its interior are of the same color, we can construct the required trees in linear time.

(5) (c) Show that if the interior of a nice triangle contains points of both colors, we can partition it into three nice triangles by connecting one of those points to all 3 vertices.

(5) (d) Given a triangle with n interior points, devise an $O(n)$ algorithm that finds an interior point p such that none of the three triangles formed by p have more than $7/8n$ points in them.

(10) (e) Show that if the convex hull of all the points is a nice triangle, we can construct the trees in $O((m+n)\log(m+n))$ deterministic time.