CS 274
Computational Geometry (Spring 2003)
Homework 5

Homework 5 is due (under my office door or in my CS mailbox) **at 5 pm on Thursday, May 22, 2003**.

Where appropriate, you may use algorithms learned in class as subroutines without re-explaining them.

[1] Overlapping intervals (9 points). Let S be a set of n intervals, and let I be a query interval.

i Given a standard interval tree T storing the intervals in S, describe an algorithm that returns all the intervals in S that intersect I in $\mathcal{O}(k + \log n)$ time, where k is the number of intervals returned.

ii How can we augment T with extra information, while still keeping its storage to $\mathcal{O}(n)$, so that we can return the *number* of intervals that intersect I in $\mathcal{O}(\log n)$ time?

iii Explain how to use a two-dimensional layered range tree to answer the same counting query, also in $\mathcal{O}(\log n)$ time. (Assume you have a correct solution to problem [5] of Homework 4; there's no need to explain that solution again.)

iv Let D be a disk, and let L be a set of l lines in the plane. Describe an algorithm that returns, in $\mathcal{O}(h + l \log l)$ time, the number h of pairs of lines that intersect in D. Hint: as best I can figure, this can be done with the range tree but not with the augmented interval tree.

[2] Voronoi diagrams (4 points). The worst-case complexity of a Voronoi diagram is $\Theta(n^2)$ in three dimensions, and $\Theta(n^{\lceil d/2 \rceil})$ in E^d.

i Prove that a planar cross-section of a d-dimensional Voronoi diagram—in other words, the planar subdivision formed by intersecting a 2-flat with a Voronoi diagram—has complexity no greater than $\mathcal{O}(n)$. Hint: What do you know about the properties of Voronoi cells?

ii Does a planar cross-section of a d-dimensional Delaunay triangulation also always have a complexity in $\mathcal{O}(n)$?

[3] Point-rectangle pairs (4 points). Let P be a set of m points in the plane, and let R be a set of n axis-parallel rectangles (possibly intersecting).

Section 10.3 of the Dutch Book includes an algorithm INSERTSEGMENTTREE that inserts an interval into a segment tree (that already contains the endpoints of the interval) in $\mathcal{O}(\log n)$ time. An interval can also be deleted in $\mathcal{O}(\log n)$ time, if a little extra information is stored. Recall that each interval is stored in linked lists in up to $\mathcal{O}(\log n)$ nodes of the tree. By using doubly-linked lists for the tree nodes, and by maintaining for each interval a list of pointers to the interval's positions in these lists, the interval can be removed from each list in constant time per list.

Describe an $\mathcal{O}(k + (m + n) \log n)$-time algorithm that reports all pairs $\langle p \in P, r \in R \rangle$ such that $p \in r$, where k is the number of such pairs.

[4] Deterministic BSPs (5 points). Give a deterministic (no randomness) algorithm that always constructs a BSP tree of size $\mathcal{O}(n \log n)$ for a set of n non-crossing line segments in the plane. Hint: Forget about auto-partitions. Use free splits wherever possible and vertical splitting lines otherwise. Expect the analysis to be a bit tricky.

[5] Minkowski sums (5 points). Suppose we want to automatically machine a two-dimensional part by cutting away the region around it with a circular drill bit. Let D be a disk (i.e. a circle with its inside filled in) whose center is the origin and whose radius is the radius of the drill bit. When machining a shape S, we may place the center of the drill bit at any point not in the interior of $S \oplus D$.

i Let P_1 and P_2 be convex polygons with nonintersecting interiors. Show that $P_1 \oplus D$ and $P_2 \oplus D$ are pseudodisks. (Note: because of the special properties of a circle, your proof should be simpler than the proof given for convex polygons in the book.)

ii If P is an n-vertex simple polygon (not necessarily convex), what is the worst-case asymptotic complexity of $P \oplus D$? (The complexity is the total number of vertices, line segments, and circular arcs needed to represent $P \oplus D$.) Briefly justify your answer. (A rigorous proof is not necessary.)

[6] Incremental visibility graphs (3 points). You are given a starting point v and a set S of polygonal obstacles in the plane. Describe how to preprocess v and S in $\mathcal{O}(n^2)$ time so that, given a query goal point w, you can return the shortest path from v to w in $\mathcal{O}(n \log n)$ time. Obviously, you will have to construct a visibility graph on S and v, but what else will you need to do?