Computational Geometry: Homework 1. Due: 1/23/2004

1. Let P_1 and P_2 be convex polygons in 2D, both represented as circular lists of vertices and n be their total number of edges. Show that there is an $O(n)$ algorithm that reports all intersections between the edges of the two polygons.

2. For P_1 and P_2 as in the previous problem, design a linear-time algorithm for computing the convex hull of $P_1 \cup P_2$, i.e. the smallest convex polygon containing both polygons.

3. Let P be any polygon (assume that the edges do not intersect), represented as a circular list of vertices. Show how to compute the convex hull of P in linear-time.

4. Assume that there is a convex hull algorithm whose running time is $O(T(n))$, where n is the number of input points. Show that using this algorithm it is possible to sort n numbers in $O(T(n) + n)$ time (note: this means that no convex hull algorithm with running time better than $O(n \log n)$ exists, since sorting requires at least that)

5. Given n line segments in the plane and a point p develop an $O(n \log n)$ algorithm that computes all segments that are visible from p (hint: sweep the plane with a rotating ray originating at p)

6. Develop an $O((n + I) \log n)$ algorithm for computing all intersections of n circles in the plane, where I is the number of intersections.

7. (this one may require a deeper understanding of the status datastructure – take a look at the 2-3 tree description in the online book) Give an algorithm that *counts* (rather than reports) intersections between n line segments, each of them being either horizontal or vertical, in $O(n \log n)$ time.