Spring Semester 2006

Computational Geometry (CS504)

This homework is due at 15:00 on **March 24**. Please put it in the homework box (see the BBS for details).

1. Consider the algorithm for fixed-radius neighbor searching we discussed in the class (see also David Mount's lecture notes). We change the size of each bucket from r to some other value, say $r/2$ or $2r$. The algorithm can easily be adapted so that it still computes the correct output. How does the running time change?

   Is it true that for buckets of size cr, where c is any constant (independent of n), the running time is still $O(n+k)$?

2. Given n points P in the plane, and a parameter $r > 0$, we want to report all pairs $p, q \in P$ where the distance between p and q is at most r. Explain how to generalize our hashing-based algorithm to do this in time $O(n+k)$. (Hint: see David Mount's lecture notes.)

3. In many situations we need to compute convex hulls of objects other than points. For instance, let S be a set of n line segments in the plane. Explain how to compute the convex hull of S in time $O(n \log n)$, and prove correctness of your algorithm.

4. Given two line segments pq and uv. Show how to test whether pq and uv intersect by doing four CCW predicate evaluations.

   Remember that $CCW(p, q, r)$ is a predicate to determine whether r lies on, to the left, or to the right of the oriented line from p to q:

$$CCW(p,q,r) = \begin{vmatrix} 1 & p_x & p_y \\ 1 & q_x & q_y \\ 1 & r_x & r_y \end{vmatrix}$$

   Note that you do not need to *compute* the intersection point, just test whether the segments intersect (so return *true* or *false*).