15-853: Algorithms in the real world — Assignment 3 (Triangulation and N-body)
Due: Thursday Nov 18, 1999

You are free to work in pairs on this assignment.
Please feel free to send me email if there are any ambiguous questions.

Problem 1: Let $\mathcal{P}$ be a non-convex polygon in 2 dimensions with n vertices. Assume it is specified by its vertices ordered around the boundary. Describe an algorithm that computes the convex hull of $\mathcal{P}$ in $O(n)$ time.

Problem 2: Show that $\Omega(n \log n)$ is a lower bound for computing the Delaunay triangulation of n points in 2 dimensions.

Problem 3: As mentioned in class, the theoretical bound on the number of interactions given by Callahan and Kosaraju for their Well-Separated Pair Decomposition algorithm is optimal asymptotically but probably very loose in terms of constant factors. To get a better feeling of what the bounds are in practice, you should implement the algorithm so that it works in 2 or 3 dimensions (it actually should not be very hard to write code that works in arbitrary dimensions). You should then run the algorithm on problem sizes ranging from 2^{10} points to 2^{18} in powers of two, using separation constants $s = 1$, 1.5, 2 and 2.5, and on the following two distributions.

1. Random points in a unit square (cube).
2. The Kuzmin distribution. This is created in 2d by generating 3d points (x_i, y_i, z_i) randomly in a unit sphere and then using the points $(x_i/z_i, y_i/z_i)$. Points can be generated in a unit sphere by generating them in a unit cube and trying again if they fall outside of the unit sphere. For 3d you can generate points (w_i, x_i, y_i, z_i) in a 4d sphere and use $(w_i/z_i, x_i/z_i, y_i/z_i)$.

You should report the number of interactions (the size of the well-separated realization) for each of the above parameters. Ideally these numbers should be plotted, but I'll accept lists of numbers. If you plot them, use one plot for each of the two distributions with size on the x-axis, number of interactions on the y-axis, and a separate curve for each s.
For information on the algorithm you can look at the class notes and/or the paper I handed out in class. You need not implement the fancy method for splitting that comes in from both sides, although you should understand why it is needed for very skewed distributions and should understand its analysis.
If you would like to try any variants of the algorithm, you are encouraged to do so.