CSC 437/537 - Geometric Algorithms

Assignment 1

Due: Tuesday, September 7th at 11:59pm

Closest Pair

1) Implement in Python a brute-force/naive algorithm for the closest pair problem. Write the code so that you can test on randomly generated point sets of various sizes/values of n. Measure the execution times for values of n large enough to see the expected asymptotic performance in a plot.
2) Implement a divide-and-conquer algorithm for the same problem. Measure its time for a similar range of values of n as in Q1 and plot to compare. Explain how your algorithm does the 'combine' or 'conquer' step and provide a geometric argument for why it works.

Convex Hull

1) Implement in Python a brute-force/naive algorithm for the 2-D convex hull problem. Write the code so that you can test on randomly generated point sets of various sizes or values of n. Measure the execution times for values of n large enough to see the expected asymptotic performance in a plot.
2) Implement an incremental construction-type algorithm such as Andrew's monotone chain convex hull algorithm for the same problem. Measure its time for a similar range of values of n as in Q1 and plot to compare.
3) Design a divide-and-conquer algorithm for the convex hull problem by doing:
   a) Let P1 and P2 be two disjoint convex polygons with n vertices in total. Give an O(n) time algorithm that computes the convex hull of P1 $\cup$ P2.
   b) Use the algorithm from part (a) to develop an O(nlogn) time divide-and-conquer algorithm to compute the convex hull of a set of n points in the plane.

Submit your code, your plots, and any screen captures that demonstrate that your programs work. For any proofs, arguments or pseudocode you can use a word processor or free-hand writing and drawing as long as it is clear and readable.