ASSIGNMENT 3

ACKNOWLEDGE YOUR SOURCES.

1. [5 marks] Consider the following randomized algorithm to find the minimum of n distinct integers. Take a random order $x_1, \ldots, x_n$ of the input numbers. Set $m = \infty$. For $i = 1 \ldots n$, if $x_i < m$ then update m to x_i.

   Use backwards analysis to argue that the expected number of times you update m is $O(\log n)$. (You may use without proof the fact that the Harmonic numbers grow as $O(\log n)$.)

2. [10 marks] This problem is about half-space range queries in the plane. Show how to pre-process a set of n points so that with space $S \in O(n)$ you can answer queries of the form, given points a and b, list all k points to the right of the line through a and b, in time $Q \in O(\log^2 n + k)$. You don't need analyze the preprocessing time. Keep your descriptions high-level. Pseudo-code is not required.

   HINT: Use the onion peeling of the point set. Suppose the layers are $H_1, \ldots H_t$ (with H_1 outermost). The main step in a query is to find the maximum index i such that H_i has a vertex v to the right of the line. Knowing v, you should be able to list all k points to the right of the line in $O(k)$ time. Of course, you may augment the onion peeling with extra information (e.g., links between successive layers) so long as the size remains $O(n)$.

3. [10 marks: 5 marks for the answer, 5 for the formatting of your reference list.]
   A literature search question. What is the best algorithm for half-space range queries in the plane? State the preprocessing time P, query time Q, and space S, and cite the original source(s). You do not need to explain the algorithm(s), but read enough to briefly answer this question: Does it appear to be practical?

   Important. Give your reference list the way research papers in CS do. Latex and a bibliography package are great for this task, although you are not required to use them. Be sure to put papers in alphabetical order by author name and to include all author names. In computational geometry (and theoretical computer science more generally) a conference paper is often followed by a more detailed journal paper. Make sure you refer to the journal paper if there is one, rather than the conference or arxiv papers. Google Scholar is an excellent tool, but if you copy citations from Google Scholar be aware that they are often wrong/incomplete, so you should check by looking at the actual paper.