Cs445 — Homework #8
Closest pair and convex hull
Due last day of class

This homework is optional — if you opt to submit it, its grade would replace the second lowest grade of your previous homeworks.

1. Run the algorithm studied in class for computing the convex hull on the "A" of the blue part of the logo of the University of Arizona. Your drawing does not have to be exact, but should be correct in terms of the order at which points are processed. Show the data structure you are using after each stage.

2. The following theorem can be proven using similar techniques to the ones we have used to prove the lower bound on sorting:

   Element Uniqueness Theorem. Let $X = \{x_1 \ldots x_n\}$ be a set of n real numbers, and let A be an algorithm that returns YES if and only if there are two elements $x_i, x_j \in X$ $(i \neq j)$ such that $x_i = x_j$. Then there is an input X for which the running time of A is $\Omega(n \log n)$.

   Show that an algorithm for finding the closest pair of points between a set of n given points in the plane runs in $\Omega(n \log n)$ time.

3. Suggest an algorithm that accepts a set $P = \{p_1 \ldots p_n\}$ and finds the three points $p_i, p_j, p_k \in P$ that the triangle that they define has the smallest perimeter among the perimeters of all $\binom{n}{3}$ triangles defined by triples of points of P. The algorithm should run in time $O(n \log n)$.

   (The perimeter of a triangle is the sum of lengthes of its edges)

4. Describe a version of the algorithm for computing the convex hull of a set of points, that use a queue, rather than a stack, and still runs in $O(n \log n)$.

5. Let P be a polygon, whose vertices $\{v_1, v_2 \ldots v_n\}$ are given to you in the order they appear along its boundary, in a clockwise order. Suggest an algorithm that finds $CH(\{v_1, v_2 \ldots v_n\})$ in time $O(n)$. Note that P does not have to be convex.

6. Let A be an array containing the vertices of a convex polygon, in a the order they appear along its boundary, in counterclockwise order. Assume that array is given to you by pointers pointing to the first and last cell in the array. Explain how to find the leftmost vertex and rightmost vertex, in time $O(\log n)$. Here n is the number of vertices.

7. Let $P = \{p_1 \ldots p_n\}$ be a given set of n points. Consider a very long corridor whose width is d. Suggest an $O(n \log n)$ time algorithm that finds whether P after being appropriatly rotation, can be translated along the corridor. Hint — compute first its convex hull.