Here are some of the questions for HW1. This is a revision of the earlier version, due by March 1. A (*) denotes a harder question. You can talk with others about the problems, but I want to insist that you write up solutions entirely on your own. If you use a source like a book, a paper, or something taken from the web, carefully cite the sources, along with your solution. If you did discuss with others, please cite them in your solutions (maybe different people for different problems). Finally, when you hand in your solutions, write the pledge “I followed the rules for this assignment” and then sign your name. If you do not do this we will not grade your HW.

1. You are given a set $S$ in $R^2$ that has $n$ distinct points $P_1, \ldots, P_n$, and for each $P_i = (x_i, y_i), x_i^2 + y_i^2 = 1$. The goal is to find a line $L$ with equation $y = cx$ that has the property that there are the same number of points of $S$ on each side of $L$.
   
   (a) Show that there always exists a line $L$ with this property.
   
   (b) Carefully describe the most efficient algorithm you can devise to find $L$, given set $S$ of $n$ points, and give its running time in terms of RAM steps. Explain why it gives a correct answer.
   
   (c) What is the best lower bound you can find for this problem? Explain why its a lower bound.

2. We are given a set $S = \{P_1, \ldots, P_n\}$ of $n$ points in general position in $R^2$ (so no three are on a line). A “polygonization” $P$ of the points in $S$ is based on a permutation $\pi = (\pi_1, \ldots, \pi_n)$ of the first $n$ integers. It has $n$ directed edges $e_i = P_{\pi_i}P_{\pi_{i+1}}, i < n$, and $e_n = P_{\pi_n}P_{\pi_1}$, and each edge is the closed, directed segment defined by its endpoints, in the given order. The polygonization $P$ is simple if no two edges meet at a point $Q$ that is interior to at least one of them. If $\pi$ defines a simple polygonization $P$ for $S$, the sequence of edges is a simple closed curve, and the Jordan curve theorem says that it divides $R^2$ into two disjoint, simply connected sets - the interior and the exterior - which are separated by the boundary sequence of edges.
   
   (a) Given a query point $Q$ carefully describe your most efficient algorithm to decide if $Q \in \text{Conv}(S)$, the convex hull of our points, and describe its running time. If the answer is “NO”, can $Q$ be inside $P$, a simple polygon based on the points of $S$? Explain your answer.
   
   (b) Now suppose we are given a permutation $\pi$ that is is claimed to define a simple polygonization of the points in $S$. Carefully describe an algorithm that can decide the veracity of this claim, and give its running time.
   
   (c) Suppose the answer above is YES. Given a query point $z = (u, v) \in R^2$ we want to know if $z$ is interior to the simple polygon (i.e., “inside, or on the boundary”). Give your most efficient algorithm for this task, explain WHY it works, and describe its complexity.
3. For $A, B \in \mathbb{R}^2$ $A \succeq^* B$ (we say $A$ “dominates” $B$) if both $A$’s $x$ and $y$ coordinates are at least as large as $B$’s. A point $P_i \in S = \{P_1, \ldots, P_n\}$ “is a maximum*” if no other point in $S$ dominates it.

(a) Give an $O(n^2)$ algorithm to find the set of maxima for a given set $S = \{P_1, \ldots, P_n\}$ of $n$ distinct points in $\mathbb{R}^2$.

(b) Now devise your most efficient algorithm ($o(n^2)$) to compute the set of maxima* of $S$. What is its complexity?

(c) Is your algorithm optimal?

(d) Discuss the problem of deciding whether a query point $Q$ is a maximum* in $S \cup Q$: (i) First assume that the inputs are $S$ and $Q$; (ii) Next assume you had already found the maxima* of $S$ as in (a) and created a useful data structures for answering the query.

(e) (*) Can you extend to three dimensions (domination is in all coordinates)?

4. In the “ultimate” planar hull algorithm, (see handout on this topic) $A$ and $B$ are points of min and max x-coordinate, respectively, $S$ is a set of $m$ points in the vertical strip defined by $A$ and $B$ and are above the line through $A$ and $B$, and $\mu$ is the median of x-coordinates in $S$. There is a unique edge (the “bridge edge”) crossing the vertical line $x = \mu$. We pair the points in $S$ and write $\sigma$ for the median of the $m/2$ slopes of the lines joining paired points. Finally $C \in S$ denotes the “support point” for the tangent line $L$ that has slope $\sigma$, has no point of $S$ above it, and has at least one point of $S$ ON the line $L$.

(a) Prove that if $C$ is to the LEFT of $x = \mu$ and if $P = (P_x, P_y)$ and $Q = (Q_x, Q_y)$ are paired and $(P_y - Q_y)/(P_x - Q_x) > \sigma$ then the point $P$ or $Q$ with SMALLER x-coordinate cannot be a Bridge vertex, and may be eliminated from the search for the bridge edge.

(b) Prove the above statement, but replacing LEFT with RIGHT, $>$ with $<$, and SMALLER with BIGGER.

(c) Explain why the above two statements allow us to “prune” at least one quarter of the $m$ points from $S$.

5. (set disjointness) $S = \{a_1, \ldots, a_n\}$ and $T = \{b_1, \ldots, b_n\}$ are two given sets each having $n$ real inputs. FIRST, carefully describe an efficient algorithm to decide (YES or NO) whether $S \cap T$ is empty. Then give an argument that establishes an $\Omega(n \log n)$ lower bound for this task based on connected components and linear decision trees. (the Edelsbrunner reference might be helpful)

6. Given $S = \{P_1, \ldots, P_n\} \subseteq \mathbb{R}^2$, the diameter diam$(S) = \max_{i<j} d(P_i, P_j)$ is the distance between the furthest pair of points. Carefully explain how to do compute it in $O(n \log n)$ RAM steps by first obtaining $CH_1(S)$ via an optimal algorithm, and then using the output to obtain the furthest pair of hull vertices in an additional $O(n)$ steps (in class we mentioned the “rotating calipers” approach for this which you might want to check out),
7. Given \( S = \{ P_1, \ldots, P_n \} \subseteq \mathbb{R}^2 \), the diameter \( \text{diam}(S) = \max_{i<j} d(P_i, P_j) \) is the distance between the furthest pair of points. In class we gave an \( \Omega(n \log n) \) lower bound for DIAMETER by reduction to set disjointness as follows:

Let \( A = \{ a_1, \ldots, a_n \} \) and \( B = \{ b_1, \ldots, b_n \} \) be the two sets whose disjointness will be tested. First, in \( O(n) \) time, we map the elements of \( A \) and \( B \) to points on the circumference of the unit circle; the elements of \( A \) will map to points in the first quadrant, and those of \( B \) will map to points in the third. To do this we first compute

\[
m = \min(a_1, \ldots, a_n, b_1, \ldots, b_n) \quad \text{and} \quad M = \max(a_1, \ldots, a_n, b_1, \ldots, b_n)
\]

and then map the sets \( A \) and \( B \) to \( \Theta = \{ \theta_1, \ldots, \theta_n \} \) and \( \Phi = \{ \phi_1, \ldots, \phi_n \} \) (respectively) using

\[
\theta_i \leftarrow \left( \frac{a_i - m}{M - m} \right) \frac{\pi}{2}; \quad \phi_i \leftarrow \pi + \left( \frac{b_i - m}{M - m} \right) \frac{\pi}{2}
\]

for \( i = 1, \ldots, n \). Finally we get the sets \( S = \{ s_1, \ldots, s_n \} \) and \( T = \{ t_1, \ldots, t_n \} \) for set-disjointness by

\[
s_i \leftarrow (\cos \theta_i, \sin \theta_i) \quad \text{and} \quad t_i \leftarrow (\cos \phi_i, \sin \phi_i), \quad i = 1, \ldots, n.
\]

(a) Show that \( A \cap B = \emptyset \iff \text{diam}(S \cup T) < 2 \).

(b) Show you can get \( S \) and \( T \) from \( A \) and \( B \) in \( O(n) \) (assume you have \( \pi \) freely available and that you can compute \( \sin \) and \( \cos \), each in constant time).

(c) Carefully argue that \( \Omega(n \log n) \) is a lower bound for diameter by reduction to set disjointness.

(d) (* for some optional extra credit, no deduction if omitted) The reduction above is flawed because it asks us to compute \( \pi, \sin, \cos \), features that are outside the unit cost RAM. Try to make the same reduction argument, but now using only operations that ARE allowed in the RAM model.

**HINT:** The key is to use trig functions and (rational) identities that relate them.

To start we map the sets \( A \) and \( B \) to angles \( \Theta' = \{ \theta'_1, \ldots, \theta'_n \} \) and \( \Phi' = \{ \phi'_1, \ldots, \phi'_n \} \) (respectively) now using

\[
\theta'_i \leftarrow 2 \cdot \arctan \left( \frac{a_i - m}{M - m} \right) \in [0, \pi/2]; \quad \phi'_i \leftarrow 2 \cdot \arctan \left( \frac{b_i - m}{M - m} \right) \in [\pi, 3\pi/2]
\]

for \( i = 1, \ldots, n \). The points \( s'_i = (\cos(\theta'_i), \sin(\theta'_i)) \) and \( t'_i = (\cos(\phi'_i), \sin(\phi'_i)) \) can be shown to be rational functions of the data and accessible in the RAM model.

Carefully explain HOW to do it, and - if you dont mind - say whether you came up with this idea or something like it all by yourself.

3