# Computational Geometry

## Assignment 1

**Date Due: Thursday, Sept 19, 2002**
**Time Due: In Class**

Your proofs should be precise, concise and clear. Assume that all polygon vertices are given in counter-clockwise order. If you need to assume general position for a set of points (that is no three points are collinear), clearly state so and briefly discuss if the assumption is important. When describing an algorithm, do so in words rather than code.

1. Let $a, b, c$ be three points in the plane that are not collinear. (a) Prove or disprove that there can be only one unique circle that has $a, b, c$ on its boundary. (b) Prove or disprove that there can be only one unique square that has $a, b, c$ on its boundary.

2. Given a set $P$ of $n$ points in the plane in general position (that is no three points are collinear), provide an algorithm that connects these points using straight line segments such that it forms a simple polygon whose $n$ vertices are the points of $P$. Explain what the running time of your algorithm is. Give an idea as to whether or not you think it is optimal. Is the general position assumption important? (that is, if we do not assume the points are in general position, can we still do this?)

3. Let $P = \{v_1, v_2, \ldots, v_n\}$ be a simple polygon.

   (a) Suppose there is a point $x \in P$ such that for every vertex $v_i$ of $P$, $[v_i x] \cap P = [v_i x]$ (this is often referred to as "x sees $v_i$"). Prove or disprove that for any point $y \in P$, $[xy] \cap P = [xy]$, i.e. $x$ sees every point in $P$.

   (b) Suppose there are two points $a, b \in P$ such that for every vertex $v_i$ of $P$, either $[v_i a] \cap P = [v_i a]$ or $[v_i b] \cap P = [v_i b]$. Prove or disprove that for any point $y \in P$, either $[ay] \cap P = [ay]$ or $[by] \cap P = [by]$, i.e. every point in $P$ is seen by either $a$ or $b$.

4. (a) Given a convex polygon $\mathcal{P}$ with $n$ vertices and a query point $q \in R^2$, carefully describe an efficient algorithm to decide if $q \in \mathcal{P}$. Can you do it in $o(n)$? Is your algorithm optimal?

   (b) Given $S = \{P_1, \ldots, P_n\}$ in $R^2$ and a query point $q \in R^2$, describe an efficient algorithm to decide if $q \in CH(S)$. Can you do it in $o(n \log n)$? Is your algorithm optimal?

5. Given a set $S$ of $n$ points, a point $p \in S$ has the *circle property* if there is a circle of finite radius through $p$ that contains $S$. Prove or disprove that the points with the circle property are exactly the vertices of $CH(S)$.