Problem Set 1
Computational and Metric Geometry

You can discuss homework problems with other students, but you must write solutions on your own. This homework is due on Monday, February 6.

Definition. The ℓ_∞ *distance* between two points $u = (u_1, u_2)$ and $v = (v_1, v_2)$ is equal to $\|u-v\| = \max(|u_1 - v_1|, |u_2 - v_2|)$. The ℓ_∞ *distance* between two sets of points $U \subset \mathbb{R}^2$ and $V \subset \mathbb{R}^2$ is

$$d(U, V) = \inf_{u \in U, v \in V} \|u - v\|_\infty = \inf_{u \in U, v \in V} \max(|u_1 - v_1|, |u_2 - v_2|).$$

Problem 1. We are given a set of n points in the plane. Design an algorithm that finds a pair of points with maximum ℓ_∞ distance in time $O(n)$. Prove the correctness of your algorithm.

Problem 2. Design an algorithm that given a set of n axis–parallel rectangles in the plane finds a pair of rectangles with minimum ℓ_∞ distance in time $O(n \log n)$. Prove the correctness of your algorithm.
Partial credit: Solve the following simpler problem for a partial credit. Design an algorithm that given a set of n axis–parallel rectangles in the plane and a parameter t finds a pair of rectangles with ℓ_∞ distance less than t in time $O(n \log n)$. If there is no such pair of rectangles, the algorithm should output that. Prove the correctness of your algorithm.

Problem 3. Design an algorithm for the following problem. The algorithm is given a set $\mathcal{C}$ of m circles and a set $\mathcal{P}$ of n points in $\mathbb{R}^2$. A circle in $\mathcal{C}$ may lie within another circle in $\mathcal{C}$, but no two circles may intersect. The algorithm must report all circles $C \in \mathcal{C}$ that contain at least one point from $\mathcal{P}$ in time $O((m + n) \ln(m + n))$. Prove the correctness of your algorithm.

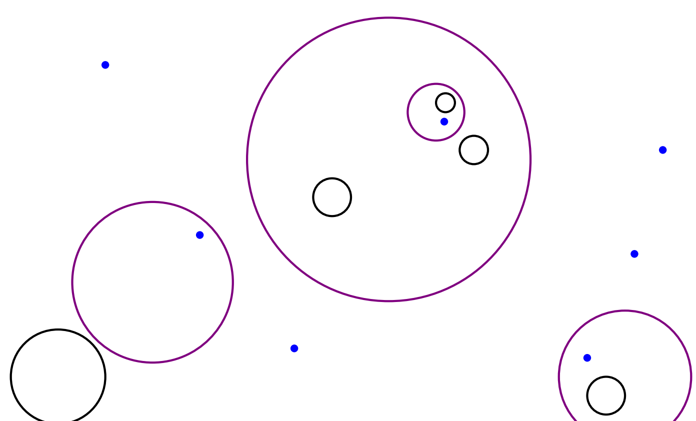

Figure 1: In this example, the algorithm must report circles shown in violet.