

Computational Geometry – Sheet 3 **Winter 2026/2027**

Prof. Dr. Sándor P. Fekete
Peter Kramer

Due 03.12.2026
Discussion 10.12.2026

Please submit your individual handwritten answers, using the box in front of `IZ 338` before *the tutorial timeslot on the due date above. Make sure to include your* full name*,* matriculation number*, and the* programme *that you are enrolled in, e.g., computer science (Master).*

You may submit your solutions in English or German. We encourage you to avoid using AI tools such as LLMs to solve these exercises.

Exercise 1 (Convex hull with lexicographical sorting). (15 points)

Given a set $\mathcal{P}$ of n points in the Euclidean plane, design an algorithm that computes the convex hull in $\mathcal{O}(n \log n)$ time, with the constraint that you may only sort the points *lexicographically*:

$$(a, b) \preceq (c, d) \Leftrightarrow \begin{cases} a < c, \text{ or} \\ a = c \text{ and } b \leq d. \end{cases}$$

You may assume that this can be done in $\mathcal{O}(n \log n)$ time, but your algorithm may not perform additional (for example, radial) sorting steps. (Hint: Try to adapt the methods of *Graham's Scan*.)

Exercise 2 (Convex hull of simple polygons). (5 points)

Given a simple polygon $P = (p_1, p_2, \ldots, p_n) \in (\mathbb{R} \times \mathbb{R})^n$ in counterclockwise order, we know that the points of P that are vertices of its convex hull are ordered identically on the hull.

Argue why this *does not* mean that we can compute the convex hull of P in $\mathcal{O}(n)$ by simply skipping the radial sorting step of *Graham's Scan* and leaving the remaining algorithm as is.

Exercise 3 (Safehouse Problem). (5+15 points)

You've successfully pulled off your biggest heist yet – you've stolen the golden Leibniz cookie! However, agents of `baked goods manufacturing co.` *are now on your tail, and all roads out of the city have been locked down in search for the famous symbol of delicacy: You cannot leave. You decide that your best bet is to find a safehouse in town as far as possible from all cookie outlets, hide the cookie there, and lie low.*

We are given an axis-aligned rectangle R in the Euclidean plane $\mathbb{R}^2$ that defines the city limits and the locations of nearby (both inside and outside the city) cookie outlets $c_1, c_2, \ldots, c_n$ as shown in Fig. 1. We are looking for a location inside R that maximizes the distance to the closest cookie outlet. That is, find $p \in R$ such that $p \coloneqq \max_{x \in R} \min_{c \in C} \|x - c\|_2$.

Assume that the Voronoi diagram $\mathrm{Vor}(C)$ is already given as a DCEL, see Fig. 2.

a) Identify a finite set of candidate locations and prove that it contains an optimal solution.

b) Design an $\mathcal{O}(n)$ time algorithm based on your candidates and prove its correctness.

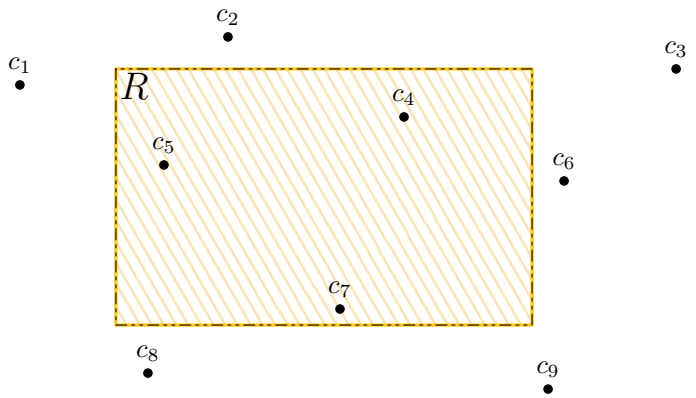


Figure 1: You require a location inside R that maximizes the distance to the nearest cookie outlet c_i.

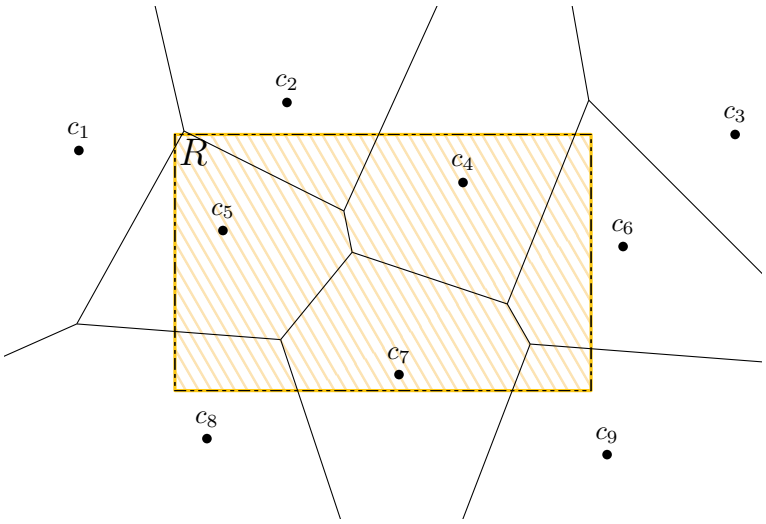


Figure 2: You can use the Voronoi diagram to guide your search.

Live Exercise 4.

Show that for a convex polygon $P = (p_1, p_2, \ldots, p_n) \in (\mathbb{R}^2)^n$ in counterclockwise order and an infinite line L through $u, v \in \mathbb{R}^2$, we can determine in $\mathcal{O}(\log n)$ time if L and P intersect.

Live Exercise 5.

Consider the setting of Exercise 3. How quickly can we solve the same problem if R is a convex polygon with m vertices?