Computational Geometry – Sheet 3 **Winter 2025/2026**

Prof. Dr. Sándor P. Fekete
Peter Kramer

Due 18.12.2025
Discussion 08.01.2026

Please submit your handwritten answers in groups of two or three*, using the box in front of* `IZ338` before *15:00 on the due date. Make sure to include your* full names, matriculation numbers*, and the* programmes *that you are enrolled in.*

In accordance with the `guidelines` *of the TU Braunschweig, using AI tools such as LLMs to solve any part of the exercises is* ***not permitted****.*

Exercise 1 (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.*

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 $c_1, c_2, \ldots, c_n$, you need a location inside R that maximizes the distance to the closest cookie outlet, see Fig. 1.

a) Identify a *(finite!)* set of candidate locations in R to choose from. Argue why an optimal solution is contained in this set. (Hint: Think about a suitable structure from the lecture.)

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

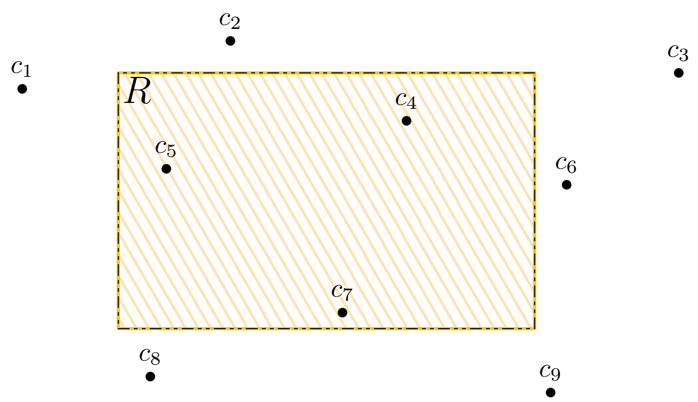

Figure 1: You require a location inside R that maximizes the distance to the nearest cookie outlet c_i.

Exercise 2 (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 3 (Convex hull of simple polygons). **(5 points)**

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

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.