Homework Exam 3 2025-2026

My name and StudentID go here!

Deadline: 14 January 2026

This homework exam has 5 questions for a total of 90 points. You can earn 10 additional points by a careful preparation of your hand-in: using a good layout, good spelling, good figures, no sloppy notation, no statements like "The algorithm runs in $n \log n$." (forgetting the $O(..)$ and forgetting to say that it concerns time), etc. Use lemmas, theorems, and figures where appropriate. Your final grade will be the number of points divided by 10. Unless stated otherwise, both randomized and deterministic solutions are allowed. In case you are asked to analyze the running time and your algorithm is randomized, analyze its expected running time.

Question 1 (13 points)

Let H be a set of $n \geq 3$ halfplanes in $\mathbb{R}^2$ that have a non-empty common intersection $I = \bigcap_{h \in H} h$. Moreover, you can assume that the bounding lines of the halfplanes in H are not all parallel. A halfplane $h \in H$ is *redundant* if it does not contribute an edge to I. Prove that for any redudant halfplane h there are two halfplanes $h_1, h_2 \in H$ for which $h_1 \cap h_2 \subset h$.

Question 2 (15 points)

Suppose that you are a shop owner. Let P be the set of n people that has entered your shop at some given day. In particular, for every person $p \in P$ you know the time a_p which he or she arrived at your shop, and the time d_p at which he or she departed. You can assume that no two people arrive or leave at the same time. You would like to store P so that given a time t and a duration ℓ, you can quickly (i.e. as fast as possible) find the number of people that were in your shop at time t and stayed for at least ℓ time units during that visit.

Develop a linear space data structure for the above problem. That is, describe your data structure, how to build and query it, and analyze the preprocessing and query time. Clearly, your data structure should have a sublinear query time (but anything strictly faster than linear time is good enough for the full points).

Question 3

Let P be a set of points in $\mathbb{R}^2$. You can assume that no three points are colinear, no two points have the same x-coordinate, and no two points have the same y-coordinate.

a. *(10 points)* Describe a data structure that given a query halfplane h can test in $O(\log n)$ time if h contains a point of P. Analyze the space usage and the preprocessing time of your data structure.

b. *(15 points)* Describe a data structure that, given an arbitrary query line segment $q = \overline{\ell r}$, with endpoint ℓ left of endpoint r, can test if there are any points in P that lie vertically below q. If, for some point p we have that $p_x \notin [\ell_x, r_x]$ it is incomparable with q (and hence it does not lie vertically below q). Analyze the space usage, the preprocessing time, and the query time of your data structure.

   Note: The number of points for this question will depend on the space usage, preprocessing time, and the query time of your data structure.

Question 4 (10 points)

Prove that incrementally constructing a Voronoi diagram on n points in $\mathbb{R}^2$ may take $\Omega(n^2)$ time. That is, give a sequence of n points $p_1, .., p_n$, such that every point p_{i+1} causes a linear number of changes (e.g. additions or removals of Voronoi vertices) in the Voronoi diagram $\text{Vor}(\{p_1, .., p_i\})$.

Question 5

Let $P = L \cup R$ be a set of n points in $\mathbb{R}^2$, where all points in L lie left of some vertical line ℓ and all points in R lie right of ℓ (we think of the points in L as "blue" and the points in R as "red"). You can assume the point set is in general position; i.e. that there are no three points on a line and that all coordinates are unique. A (point on) an edge e of $\text{Vor}(P)$ is *bichromatic* if and only if exactly one point defining e lies in L (and thus the other point lies in R).

a. *(8 points)* Prove that any horizontal line h contains one bichromatic point.

b. *(10 points)* Let $B \subset \mathbb{R}^2$ be the set of bichromatic points. Prove that B forms an unbounded y-monotone polygonal chain with $O(n)$ edges.

c. *(9 points)* Describe how to efficiently compute B, and how this leads to an $O(n \log n)$ time algorithm to compute the Voronoi diagram $\text{Vor}(P)$.

   Note: Your solution is supposed to be an alternative to the algorithm from Theorem 7.10 in the book. Clearly, you are not allowed to just directly use that result to construct $\text{Vor}(P)$. Instead, your solution should actually use B.