Homework Exam 3 2024-2025

My name and StudentID go here!

Deadline: 15 January 2023, 13:15

This homework exam has 6 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

a. *(10 points)* Let P be a set of n points in $\mathbb{R}^2$. Prove that the average degree of any triangulation on P is at most 6.

b. *(10 points)* Prove that for any n there exists a set P of n points in $\mathbb{R}^2$ so that any triangulation of P has a vertex of degree $n-1$.

Question 2 (15 points)

Suppose that we have a trace $\mathcal{T}$ that records the n operations that an algorithm executes on a (dynamic) dictionary storing some dynamic set X of real numbers. That is, every entry in $\mathcal{T}$ is a triple consisting of a time stamp t, an operation, which is either INSERT or DELETE, and a value $v \in \mathbb{R}$ –the number that is inserted or deleted. Describe a data structure that allows us to efficiently "replay" the queries of the algorithm. That is; given an arbitrary query pair (t, q) consisting of a time $t \in \mathbb{R}$ and a value $q \in \mathbb{R}$ it allows us to efficiently report the value v that was the successor of q at time t. Analyze the space, preprocessing time, and query time of your solution.

The number of points awarded will depend on the space, preprocessing, and query time of your solution.

Question 3 (10 points)

Let $\mathcal{S}$ be a set of n axis-parallel squares. Develop a data structure that can store $\mathcal{S}$ that can answer the following queries: given a point q, report a largest square $S^* \in \mathcal{S}$ that contains q. Argue/prove that your data structure answers queries correctly. Aim for the fastest queries possible, while using $O(n \log^c n)$ space (for some constant c).

The number of points awarded will depend on the query time and the space used by your data structure.

Question 4 (10 points)

Let $\mathcal{T}$ be a kD-tree on a set P of n points in $\mathbb{R}^2$, in which each node has been annotated with the number of points in its subtree. Analyze the worst case query time for a range counting query on $\mathcal{T}$ with a query disk D. Argue that your analysis is tight in the worst case.

Question 5

Let P be a set of n points in $\mathbb{R}^2$, and let $NN(p)$ denote the (Euclidean) *nearest-neighbor* of p. That is, $NN(p) = \operatorname{argmin}_{q \in P} \|pq\|$.

a. *(10 points)* Prove that the Voronoi regions of p and $NN(p)$ are adjacent in the Voronoi diagram $VD(P)$ of P.

b. *(10 points)* Describe an $O(n \log n)$ time algorithm to compute, for every point $p \in P$, its nearest neighbor $NN(p)$.

Question 6 (15 points)

Let R be a set of n "red" points in $\mathbb{R}^2$, let B be a set of n "blue" points in $\mathbb{R}^2$, let $age : R \cup B \to \mathbb{R}$ be a function that assigns an age to every point, and let $\Delta > 0$ be some real number. You can again assume that all coordinates and ages are unique, and that there are no three colinear or four corcicular points. Design an algorithm to compute, for every red point $r \in R$, the closest (in terms of the Euclidean distance) blue point b among B for which $age(b) \in [age(r), age(r) + \Delta]$. Your algorithm should run in subquadratic time.