Retake Homework Exam 2019

Deadline: 13 March 2020, 17:00

This homework exam has 5 questions for a total of 90 points. You can earn an additional 10 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 (with a maximum of a 10).

Question 1 *(10 points)*

Let P be a set of n points in $\mathbb{R}^3$, and let h be a real number. You can assume that all coordinates are unique. Consider a horizontal slab S of height h (i.e. $S = \{(x, y, z_{\min} + z') \mid x, y \in \mathbb{R}, z' \in [0, h]\}$ for some $z_{\min} \in \mathbb{R}$), and let $V(S)$ denote the volume of the axis-aligned bounding box of $P \cap S$. Develop an $O(n \log n)$ time algorithm that can compute the maximum value of $V(S)$ over all horizontal slabs S of height h. Argue that your algorithm is correct and achieves the desired running time.

Question 2

Let $\mathcal{S}$ be a planar subdivision with n vertices, represented as a DCEL.

(a) *(10 points)*
Give pseudo-code for an algorithm that, given a pointer to a face f, computes *the set* of all faces of $\mathcal{S}$ adjacent to f. Your algorithm should use the 'Twin', 'NextEdge', 'PrevEdge', etc. fields to navigate (i.e. you cannot assume you can directly access a list of vertices). Describe in a few sentences the main idea of your algorithm. Furthermore, argue that it is correct.

(b) *(5 points)*
Analyze the running time of your algorithm. In particular, argue whether or not it is output sensitive, i.e. if its running time depends on k the number of faces reported.

Question 3

A simple polygon P is *star-shaped* if and only if it contains a point $q \in P$ such that for any point $p \in P$, the line segment $\overline{pq}$ lies inside P.

(a) *(10 points)*
We say that P is *strictly star-shaped* if (and only if) the point q is also a vertex of P. Show that there are polygons that are star-shaped but not strictly star-shaped.

(b) *(10 points)*
Design an algorithm that can decide if a simple polygon P with n vertices is star-shaped in $O(n)$ expected time. You can again assume no three vertices of P are colinear.

Question 4 *(20 points)*

Let S be a set of n line segments in $\mathbb{R}^2$. You can assume no three endpoints in S are colinear, and all coordinate values of all endpoints are unique. Describe an $O(n^2)$ time algorithm to compute the maximum number of segments from S that can be stabbed by a line. Your algorithm should report this number, k, and a line ℓ realizing k intersections.

Question 5

Let P be a polyline with n vertices. You can assume that P does not have any self intersections, that all coordinates of the vertices are unique, and that no three vertices are colinear.

(a) *(5 points)*
Describe a data structure that can efficiently test if a query point $q \in \mathbb{R}^2$ lies on an edge of P, and if so, returns this edge. In particular, your data structure should answer these queries in worst case $O(\log n)$ time. Briefly analyze the preprocessing and space required.

(b) *(7 points)*

Given two points $s, t \in \mathbb{R}^2$, let $P[s, t] \subseteq P$ be the maximal polyline with starting point s and ending point t (and note that $P[s, t] = \emptyset$ if s or t does not lie on P).

Describe a data structure that stores P and can answer the following queries in $O(\log^c n)$ (expected or worst case) time, for some $c > 1$: given two query points $s, t \in \mathbb{R}^2$ report the axis-parallel bounding box of $P[s, t]$.

Analyze the space usage of your data structure and its preprocessing time.

The number of points awarded for this question depends on the query time and space usage of your data structure. Aim for the fastest possible queries, while using subquadratic space.

(c) *(10 points)*

Describe a data structure that stores P and can answer the following queries in $O(\log^c n)$ (expected or worst case) time, for some $c > 1$: given three points $s, t, q \in \mathbb{R}^2$, report the vertex in $P[s, t]$ closest to q. Analyze the space usage of your data structure and its preprocessing time.

The number of points awarded for this question depends on the query time and space usage of your data structure. Aim for the fastest possible queries, while using subquadratic space.

(d) *(3 points)*

Suppose that P may have self intersections. Briefly describe why/where in your solution for question c you need that P has no self-intersections.