Homework Exam 3 2019

Deadline: 10 January 2020, 12:45

This homework exam has 4 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

Let P be a simple polygon with n edges in which no two vertices have exactly the same x or y-coordinate, and no three points are colinear. We wish to preprocess P for efficient point location queries. Clearly, we can just build a trapezoidal decomposition on the edges of P in expected $O(n \log n)$ time. Here, we will show that we can do slightly better by using a variation of the randomized incremental approach we discussed in class.

Consider the following approach:

split the boundary of P into $\lceil n/m \rceil$ pieces with m consecutive edges each. Consider these pieces in a random order, and for each of the n/m pieces do the following:

1. do a single point location query to find the trapezoid containing the starting point of the piece,
2. trace the piece in the decomposition to find all existing trapezoids intersected by the piece,
3. delete all these trapezoids and replace them by new trapezoids.

Analyze the above algorithm. In particular:

(a) *(10 points)*
Prove that in total the expected number of trapezoids created is still $O(n)$.

(b) *(5 points)*
Prove that the total expected time it takes to find all trapezoids that we have to delete is $O(nm)$.

(c) *(10 points)*
Argue that in expected $O(n\sqrt{\log n})$ time we can build a point location data structure that has expected size $O(n)$ and can answer point location queries in expected $O(\log n)$ time.

Question 2 *(20 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.

Clarification 2019-12-20: Your data structure should have a sublinear (i.e. $o(n)$) query time.

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, and preprocessing time of your data structure.

(b) *(10 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).

Question 4

Let P be a set of n weighted points in $\mathbb{R}^2$, that is, every points $p \in P$ has a weight $p_w \in \mathbb{R}$. Furthermore, let $\|ab\|$ denote the Euclidean distance between to points $a, b \in \mathbb{R}^2$.

(a) *(15 points)*
We can define a distance function $d^+(p, q) = p_w + \|pq\|$ that measures the "weighted" distance from any point $q \in \mathbb{R}^2$ to a point $p \in P$, and thus we can define a Voronoi diagram based on the distance measure d^+. Prove that the Voronoi region $V^+(p)$ of p (i.e. the set of all points in $\mathbb{R}^2$ closer to p than to any other site $z \neq p \in P$) is connected.

(b) *(10 points)*
We can also define the distance function $d^*(p, q) = p_w\|pq\|$. Prove that in this case, the Voronoi region $V^*(p)$ of p (i.e. the set of all points in $\mathbb{R}^2$ closer to p than to any other site $z \neq p \in P$) may be disconnected.