Final Exam 2018-2019

30 January 2019, 17:00-20:00

This exam has 5 questions for a total of 90 points. You can earn an additional 10 points if you write readable, unambiguous, and technically correct. No statements like "The algorithm runs in $n \log n$." (forgetting the $O(..)$ and forgetting to say that it concerns time), etc. Your final grade will be the number of points divided by 10.

Read every question carefully (!), make sure you understand it, and be sure to answer the question. Answer questions in sufficient but not too much detail. You may *not* use the textbook, or any other notes during the exam. Be sure to put your name on every piece of paper you hand in. Good Luck!

Question 1 *(15 points)*

For each of the following tasks, state the worst case (or expected, if the algorithm is randomized) running time for the best possible algorithm to perform the task. Use k to denote the output size if applicable.

(a) Computing the smallest enclosing ball of n points in $\mathbb{R}^3$.
(b) Computing the convex hull of an x-monotone polygonal chain with n vertices.
(c) Querying a KD-tree on n points in $\mathbb{R}^3$ to report all points in an axis-aligned query box $[x_1, x_2] \times [y_1, y_2] \times [z_1, z_3]$.
(d) Computing an Euclidean Minimum Spanning tree of n points in $\mathbb{R}^2$.
(e) Inserting a new line into an arrangement of n lines (in $\mathbb{R}^2$).
(f) Constructing a data structure on a set P of n points in $\mathbb{R}^2$ that can count the number of points of P in an arbitrary query triangle in $O(\log^3 n)$ time.
(g) Triangulating all bounded faces in an arrangement of n lines (in $\mathbb{R}^2$).

Question 2

Interval trees and segment trees can both be used to store n intervals in $\mathbb{R}^1$ so that we can efficiently report all intervals containing a query value $q \in \mathbb{R}^1$.

(a) *(6 points)*
Describe both data structures in a few lines each and give their preprocessing times, space usage, and query times.

(b) *(4 points)*
Sketch a situation in which you would want to prefer to use an interval tree rather than a segment tree and vice versa.

Question 3 *(10 points)*

Let $\overline{pq}$ be a vertical line segment with endpoint p above endpoint q, and let r be a point whose x-coordinate is larger than that of q and whose y-coordinate is larger than that of p. The line ℓ through r intersects $\overline{pq}$ in a point s.

Formulate the above paragraph in its dual form using the usual point-line duality. It does not have to be a literal translation, but it should capture all geometric information from the above paragraph. Draw the construction in the dual plane, and clearly label all of the above objects, $\overline{pq}^*, p^*, q^*, r^*, s^*, \ell^*$, in your drawing.

See the other side for the remaining questions.

Question 4

Let P be a set of n points in $\mathbb{R}^2$, let CH denote the convex hull of P, and let VD denote the Voronoi diagram of P.

(a) *(10 points)*
Prove that if p occurs as a vertex of CH then the Voronoi cell of p in VD is unbounded.

(b) *(10 points)*
Assume that P is in general position; i.e. there are no four points that are cocircular and no three colinear points. Let k be the number of points on CH. Prove that the number of vertices in VD is exactly $2n - 2 - k$.

Question 5

Let $\mathcal{S}$ be a planar subdivision that has n edges, represented by line segments, and let P be a set of m points in $\mathbb{R}^2$.

(a) *(4 points)*
Describe a data structure that uses $O(n)$ expected space and that given a query point $q \in \mathbb{R}^2$ allows us to find the face of $\mathcal{S}$ containing q in $O(\log n)$ expected time.

(b) *(10 points)*
Prove that the expected query time of the data structure you described is indeed $O(\log n)$ using backward analysis.

(c) *(2 points)*
Suppose that we want to compute, for every point in P, the face of $\mathcal{S}$ that contains it, using the data structure above. What is the total expected running time of this approach?

(d) *(15 points)*
Describe a different approach to solve the problem from question (c) (i.e.: for every point in P, find the face of $\mathcal{S}$ containing it) that achieves the same running time, but now in the worst case. Briefly argue that your algorithm is correct and achieves the stated running time.

(e) *(4 points)*
Suppose that you would want to implement either of these two solutions. State two potential difficulties that you have to deal with typical to geometric algorithms. Moreover, for both difficulties give a possible solution.