Final Exam 2021-2022

02 February 2022, 17:00-20:00

This exam has 8 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 (10 points)

For each of the following tasks, state the running time for the best possible algorithm to perform the task. If the algorithm is deterministic, give the worst case running time. If the algorithm is randomized, indicate this and give the expected running time. Use k to denote the output size if applicable.
Stating only the running time is sufficient, no need to explain your answers in detail.

(a) Triangulating a y-monotone polygon with n vertices.
(b) Given a kd-tree built on a set of n points in $\mathbb{R}^3$. Reporting all points in an axis parallel query box.
(c) Given a planar subdivision $\mathcal{S}$ (as a DCEL) with n vertices, preprocessing $\mathcal{S}$ into a data structure that can answer point location queries in expected $O(\log n)$ time.
(d) Computing the convex hull of an x-monotone polygonal chain with n vertices.
(e) Given a planar subdivision $\mathcal{S}$ (as a DCEL) with n vertices, and a pointer to a face F in $\mathcal{S}$, reporting all edges of F.

Question 2 (5 points)

Which fields does a half-edge in a DCEL store?

Question 3 (9 points)

List three reasons/applications why you may want to compute the Delaunay triangulation of a set of n points.

Question 4 (6 points)

Argue why range trees are more similar to segment trees than to interval trees.

Question 5 (10 points)

Let S be a set of n disjoint line segments in $\mathbb{R}^2$, and let T be the vertical decomposition of S. Prove that the total complexity of T (i.e. the total number of vertices, edges, and faces) is $O(n)$.

Question 6 (15 points)

Let $\overline{pq}$ be a line segment with endpoints p and q, with q to the top-right of p. Let ℓ and m be two lines that intersect in a point r. Line m is horizontal, intersects $\overline{pq}$, and the intersection point of ℓ and m lies to the right of $\overline{pq}$. Furthermore, the slope of ℓ is bigger than that of the supporting line of $\overline{pq}$.

(a) Draw the above construction in the primal and in the dual plane. Clearly label the following objects and their duals in your drawing: $\overline{pq}, p, q, r, \ell, m$.
(b) 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.

Question 7

Let $\mathcal{S}$ be a set of n axis-parallel unit squares, and let $\mathcal{U} = \bigcup_{S \in \mathcal{S}} S$ be the union of these unit squares.

(a) *(10 points)* Prove that the boundary of $\mathcal{U}$ consists of at most $O(n)$ edges.

(b) *(8 points)* Does the above $O(n)$ bound on the complexity of (the boundary of) $\mathcal{U}$ still hold when the regions in $\mathcal{S}$ are axis-parallel rectangles rather than axis parallel unit squares? Briefly (at most one paragraph) argue why or why not.

Question 8

Let P be a set of n point sites in $\mathbb{R}^2$ and recall that the k^{th}-order Voronoi diagram $VD_k(P)$ is a subdivision of $\mathbb{R}^2$ into maximal cells, each of which corresponds to the set of k-closest sites. More precisely, let a and b be any two points in a cell F of $VD_k(P)$, let $p_1, .., p_n$ be the points in P ordered by increasing distance from a, and let $q_1, .., q_n$ be the points in order of increasing distance from b, then $p_1, .., p_k = q_1, .., q_k$.

In this question, we will develop a (hopefully) simple $O(n^4)$ time algorithm to compute $VD_k(P)$.

(a) *(5 points)* Characterize the cells of $VD_k(P)$, i.e. briefly state what properties they have. (Informally: "Describe, what the cells look like").

(b) *(6 points)* Use the properties that you discovered in question (a) to argue that we can compute $VD_k(P)$ in $O(n^5 \log n)$ time. There is no need to give a full correctness proof; just state the main ideas/arguments in one or two paragraphs.

(c) *(6 points)* Argue that we can actually implement the algorithm from question (b) to run in $O(n^4)$ time.

Note that you only have to argue that you can compute the subdivision $VD_k(P)$. There is no need to worry about how to actually find or retrieve the k sites corresponding to a particular cell of $VD_k(P)$.