Final Exam Geometric Algorithms, January 30, 2009, 9–12

Read every question carefully, make sure you understand it, and be sure to answer the question. Read the question again after answering it, as a check whether you really answered the question. Answer the easier questions first, and then the harder ones. You may not use any algorithmic result from the book, unless it is explicitly stated that this is allowed in the question. Answer questions in sufficient but not too much detail. You may not use the textbook during the exam.

Be sure to put your name on every piece of paper you hand in. Also write down your "collegekaartnummer". If you write readable and unambiguous enough, you get one point for free. The other nine points can be earned by answering the questions correctly. Good luck!

1. (1 point) Given four points $p_1 = (1,1)$, $p_2 = (1,3)$, $p_3 = (2,2)$, and $p_4 = (3,3)$. Consider the bisector of the line segments $s = \overline{p_1p_2}$ and $s' = \overline{p_3p_4}$. Draw the line segments s and s' together with their bisector. Give the coordinates of the vertices of the bisector.

2. (1 point) Is it true that for any subdivision and any half-edge $\vec{e}$, we have

   (a) $\mathit{IncidentFace}(\vec{e}) = \mathit{IncidentFace}(\mathit{Next}(\vec{e}))$

   (b) $\mathit{Origin}(\vec{e}) = \mathit{Origin}(\mathit{Next}(\mathit{Twin}(\vec{e})))$

   (c) Suppose that a subdivision S is given where for every edge e, it is the case that either $\mathit{Next}(\vec{e}) = \mathit{Twin}(\vec{e})$, or $\vec{e} = \mathit{Next}(\mathit{Twin}(\vec{e}))$ (but not both). Describe in words what such the subdivision S looks like.

3. (1 point) The dual of a (non-vertical) line segment is a double wedge. More precisely, the collection of all lines that intersect a line segment dualizes to the collection of all points inside the double wedge that is the dual of that line segment.

   A double wedge has a left part and a right part. Why does it have two parts? Does it always have two parts? How do the two parts relate to the lines that intersect the line segment that was dualized to the double wedge?

4. (1 point) Let L be a set of n lines in the plane. Assume that no line of L is vertical and no two lines of L are parallel. Give an $O(n \log n)$ time algorithm that takes L as its input and computes an axis-parallel rectangle that contains all vertices of the arrangement $\mathcal{A}(L)$ in its interior.

5. (1.5 point) Given a set S of n point sites in the plane. We want to preprocess S into a data structure that can answer queries of the following type efficiently: For a vertical query line segment q, report all sites that are the closest site for at least some point on q.

Explain what data structure you would use, how you would construct it, what the preprocessing time and storage requirements are, and what the query time is (ignore any degenerate cases). You may refer to and use results from the book freely, but refer to the book whenever you do.

6. (1.5 point) When we use randomized incremental construction to construct the Delaunay triangulation, we must analyze how many flips are done in the expected case, using backwards analysis. Assume we have the Delaunay triangulation D_{i-1} of the points $p_{-2}, p_{-1}, p_0, \ldots, p_{i-1}$, we add the next point p_i, and construct the new Delaunay triangulation D_i of the points $p_{-2}, p_{-1}, p_0, \ldots, p_i$.

   When we locate p_i inside a triangle of D_{i-1}, we first connect p_i to the three vertices of that triangle. Then we restore the Delaunay property by flipping. Prove using backwards analysis that the expected number of flips needed to make D_i is at most three.

7. (2 points) Let S be a set of n triangles in the plane. Their boundaries are disjoint but it is possible that one triangle lies completely inside another triangle. Let P be a set of m points in the plane. Give a plane sweep algorithm that reports all points of P that lie outside all triangles of S.

   Give the status, describe the event handling, and analyze and state the running time (expressing your time bound in n and m). You may ignore all degenerate cases.