

Final Exam Geometric Algorithms, April 21, 2011, 14–17

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 “studentnummer”. If you write readable, unambiguous, and technically correct, you get one point for free (in particular, do not write “line” if you mean “line segment”, do not write “wedge” when you mean “double wedge”, and do not write “Step 1 takes $n \log n$ time” when you mean “Step 1 takes $O(n \log n)$ time”). The other nine points can be earned by answering the questions correctly. Good luck!

1. (0.5 point) The Delaunay triangulation of a set of points in the plane optimizes some geometric, measurable attribute over all possible triangulations of the point set. What is the thing that the Delaunay triangulation optimizes? Formulate your answer precisely.
2. (1 point) Explain how a Voronoi diagram of line segments in the plane can be used for motion planning purposes. Explain this in about 50–100 words.
3. (1 point) What are the ways to deal with degenerate cases in geometric algorithms? Explain these briefly.
4. (1 point) In the plane, there are two line segments s_1 and s_2 . These two line segments are separated by a vertical strip. Furthermore, the supporting line ℓ of s_1 (so ℓ contains s_1) intersects the line segment s_2 .

Formulate the paragraph above with statements in its dual setting (not literally but in all aspects of its meaning), and draw the dual picture, where s_1^* , s_2^* , and ℓ^* are indicated.

5. (1.5 points) Let S be a set of n disjoint line segments in the plane, and let p be some point that does not lie on any line segment of S . Line segments are closed, that is, they include their endpoints.

Describe a sweep-based algorithm that computes all line segments of S that are visible from p (that is, it is not possible from p to look through a line segment of S and see anything behind it). You should report all line segments of S of which at least some point is visible.

Please turn over

6. (2 points) We wish to store a set R of n rectangles in the plane in a data structure for rectangle intersection queries. All rectangles are axis-parallel and include their interior.
 - (a.) Analyze the different ways in which two rectangles can intersect, and formulate which query problems need to be solved together to answer rectangle intersection queries.
 - (b.) Explain the data structures that solve the query problems formulated under (a.) briefly, and give the storage and query time performance of each.
7. (2 points) Let L be a set of n non-vertical lines in the plane. We will build an arrangement A of some subset of the lines of L inside a big, given rectangular region, using the following algorithm:

1. Put the lines of L in random order, giving ℓ_1, \dots, ℓ_n .
2. Let p_1 be the intersection point of the y -axis and ℓ_1 .
3. Make an arrangement A with line ℓ_1 only, inside a big, given rectangle.
4. FOR $i \leftarrow 2$ TO n :
5. IF the line ℓ_i passes strictly below p_{i-1} ,
6. THEN insert ℓ_i into the arrangement A
7. $p_i \leftarrow$ the intersection point of ℓ_i and the y -axis
8. ELSE $p_i \leftarrow p_{i-1}$
9. RETURN A

Insertion of a line into an arrangement of lines is done in the normal way, as treated in Chapter 8 of the book.

- (a.) What is the worst-case running time of the algorithm? Prove your answer. You may use results from the book if you clearly indicate what result you are using and where.
- (b.) What is the expected size of the arrangement that is built? Prove your answer.
- (c.) What is the expected running time of the algorithm? Prove your answer.