

Final Exam Geometric Algorithms, April 16, 2015, 17.00–20.00

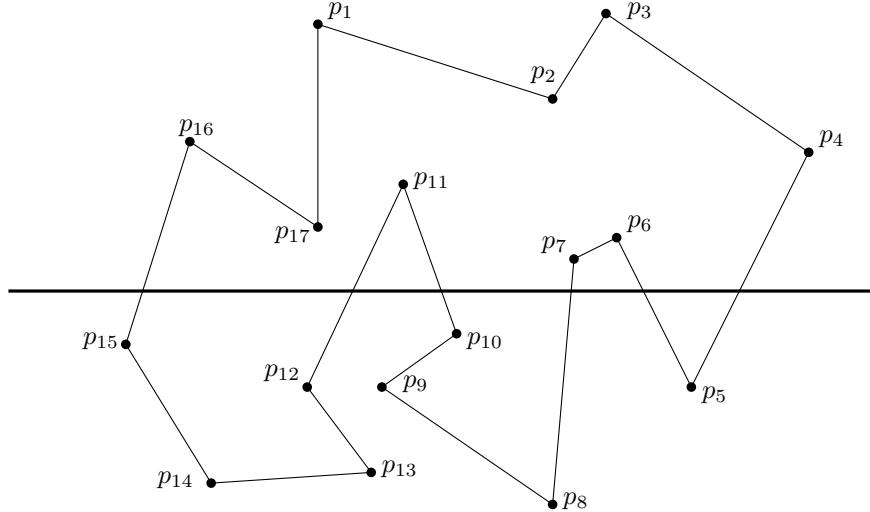
Read every question carefully, make sure you understand it, and be sure *to answer the question*. Read the question again after answering it, as a check whether you really answered the question. Answer the easier questions first, and then the harder ones. Answer questions in sufficient but not too much 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” 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. (2 points; give short answers; -0.5 point for every wrong or missing answer)
 - (a.) What is the query time for a range reporting query using a range tree for a set of n points in the plane? Give the query time both without and with fractional cascading.
 - (b.) How is an α -shape defined?
 - (c.) Suppose you want to show that problem X is NP-hard, and you already know that problem Y is NP-hard. Describe what you need to do to prove that problem X is NP-hard, using a *reduction*.
 - (d.) Explain why geometric algorithms often assume that the input is in *general position*.
 - (e.) Suppose we have a set of 21 line segments in the Euclidean plane. All segments have different endpoints, one line segment intersects all 20 others, and these 20 other line segments each intersect 5 line segments. All intersection points are proper; exactly two line segments pass through any intersection point. What is the number of faces of the resulting subdivision (including the unbounded one)?
2. (1 point)
 - (a.) Given a DCEL of a subdivision consisting of n vertices in a chain with $n - 1$ edges. Is it possible to remove a particular edge from this subdivision, and restore the DCEL to a correct DCEL in $O(1)$ time in all cases? Explain how, or why not. The edge to be deleted is given by a pointer to the representation to a half-edge of that edge in the DCEL.
 - (b.) Given a DCEL of a subdivision consisting of n vertices in a cycle with n edges, so it is a subdivision with one simple polygon, the interior face and the exterior face. Is it possible to remove a particular edge from this subdivision, and restore the DCEL to a correct DCEL in $O(1)$ time in all cases? Explain how, or why not. The edge to be deleted is given by a pointer to the representation to a half-edge of that edge in the DCEL.
3. (1 point)

The planar point location data structure is built by a randomized incremental construction algorithm. To analyze the size of the data structure we use backwards analysis. Give the backwards analysis argument explicitly and continue to prove that the size of the data structure is $O(n)$ expected, where n is the number of line segments on which the structure is built.

4. (1 point)



In the polygon triangulation algorithm, first phase, a *helper* was used with the edges in the status structure, to help decide to which vertex a split vertex must be connected to make the polygon y-monotone. In the figure, edges $\overline{p_5p_6}$, $\overline{p_{10}p_{11}}$ and $\overline{p_{15}p_{16}}$ have a helper.

- (a.) Specify for each of these edges which vertex is the helper, when the sweep-line is at the shown position in the figure.
- (b.) What are the actions performed by the algorithm when the sweep-line contains p_6 , and this event is handled?

5. (1.5 point)

Let P be a set of n points in the plane. We want to store the points in a data structure for triangle range queries, where a triangle is bounded by a horizontal edge, then (clockwise) a slope $+1$ edge, and then a slope -1 edge. Describe how a kd-tree can be used for this type of query. Give the resulting query time and size of the data structure.

Hint: Think about one of the recommended exercises in the book.

6. (1.5 point)

Given a set of n non-vertical line segments in the plane (which may intersect), give an algorithm to decide if a non-vertical line exists that intersects all of the line segments. To this end:

- (a.) Phrase the problem as a proper computational geometric problem in its dual setting.
- (b.) Give the algorithm by sketching the main steps. The algorithm should run in $O(n^2)$ time. Also analyze the running time of your algorithm.

7. (1 point + 0.5 bonus point)

For any integer $k \geq 3$, let P_k be a set of $k+1$ points, consisting of the center and the k vertices of a regular k -gon. Let V_k be the (unbounded) Voronoi diagram of P_k .

- (a) Draw V_4 and V_6 . You are allowed to draw half-lines only partially.
- (b) Compute the dilation of V_4 and V_6 .
- (c) For which value of $k \geq 3$ does V_k have the smallest dilation?