

Final Exam Geometric Algorithms, February 1, 2007, 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. Make 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 book during the exam.

Be sure to put your name on every piece of paper you hand in. Also write down your “collegekaartnummer”. Good luck!

1. (1 point) Consider the following two very similar problems.
 - (a) Let H be a set of horizontal line segments in the plane, and suppose we wish to determine a line that intersects all of them.
Describe the dual version of this problem, where you capture all aspects of the original problem specification in the dual version.
 - (b) Let V be a set of vertical line segments in the plane, and suppose we wish to determine a line that intersects all of them.
Describe the dual version of this problem, where you capture all aspects of the original problem specification in the dual version.
2. (2 points) Let S be a set of n disjoint non-vertical line segments in the plane.
 - (a) Describe a data structure requiring $O(n \log n)$ storage that allows us to determine those segments in S that intersect a ray running vertically upward from a query point (q_x, q_y) in $O(\log n + k)$ time, where k is the number of reported answers. (Hint: adjust the associated structures of the segment tree to achieve the required query time.) Justify the correctness of your solution.
 - (b) Describe a data structure requiring expected linear storage that allows us to determine the *first* segment in S intersected by a ray running vertically upward from a query point (q_x, q_y) in $O(\log n)$ expected time. (Hint: consider for which points (q_x, q_y) the reported segment will be the same.) Justify the correctness of your solution.
3. (1 point) Describe algorithms to insert and delete points from a 2-dimensional range tree. You don’t have to take care of rebalancing the structure.
4. (2 points) Let P be a set of n points in the plane, and let $p \in P$ be one of these points. Give a randomized algorithm that can decide in $O(n)$ expected time whether p is a vertex of the convex hull of P . Give the algorithm and the running time analysis. (You may give the algorithm in a precise textual description; you need not use pseudo-code.)

5. (1 point)

- (a) Construct the Minkowski sum of the square P with vertices $(1, 1)$, $(-1, 1)$, $(-1, -1)$ and $(1, -1)$ and the square Q with vertices $(1, 0)$, $(0, 1)$, $(-1, 0)$ and $(0, -1)$. Give your answer by specifying the coordinates of the output in counterclockwise order.
- (b) Use the bound on the complexity of the Minkowski sum of two convex polygons to prove that the complexity of the Minkowski sum of a convex polygon P with m vertices and a non-convex polygon Q with n vertices is $O(mn)$.

6. (1 point) Let P be a set of 200 points in the plane. No three points of P lie on a straight line, and no four points of P lie on a circle. Suppose that the Voronoi diagram of P has 360 Voronoi vertices. How many points of P lie on the convex hull? Explain briefly how you obtained your answer.

7. (2 points) Let S be a set of n unit squares in the plane, and let P be a set of n points. Give an efficient plane sweep algorithm that determines the number of points from P that are not contained in any of the squares from S .

Define the status, describe the status structure, define the events that your algorithm will have, and how they are treated.

Analyze the time taken for handling each event, and use this to analyze the overall running time of the algorithm.