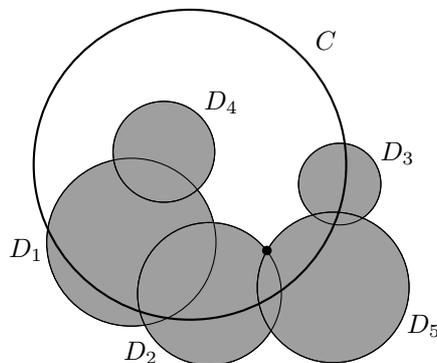


Final Exam Geometric Algorithms, February 1, 2008, 9–12

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. 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 textbook 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 point) Explain the query algorithm for a query point q in a normal interval tree that stores a set S of intervals. (Do not confuse interval trees and segment trees.)
- (a.) (1 point) Given a set P of n points in the plane, the *diameter* of P is the maximum distance that occurs between a pair of points in P . Suppose we want to compute the pair of points of P that realizes the diameter. Why can't we solve this problem in linear expected time using randomized incremental construction (in the manner of linear programming, or smallest enclosing discs)?
(b.) (1 point) Given a circle C and a set D of discs that all intersect C or lie interior to it. Suppose we wish to compute the lowest point inside C or on its boundary that is not inside any disc of D . In the figure below, the upper intersection point of the boundary of D_2 and D_5 is the solution (indicated).



Why can't this problem be solved using randomized incremental construction in linear expected time (in the manner of linear programming, or smallest enclosing discs)?

- (1 point) Two line segments s_1 and s_2 intersect if the line ℓ_1 containing s_1 intersects s_2 and the line ℓ_2 containing s_2 intersects s_1 .

Translate this geometric fact into a dual setting (thereby describing what it means in the dual that two line segments intersect). You may assume that s_1 and s_2 are not vertical.

4. (2 points) Some person Xyz wants to develop a data structure that can store a set P of n points in the plane, so that for any query point, the closest two points of P can be reported efficiently. Person Xyz takes the following approach. First, build the Voronoi diagram of P , and preprocess its edges into a planar point location structure. This allows us to find the closest point of P to any query point q efficiently.

All trapezoids that are part of the planar point location structure will not only store the point of the Voronoi cell they belong to, but also an extra pointer to a data structure. Let p_i be any point of P . Then all trapezoids in the Voronoi cell of p_i have a pointer to a data structure T_i . This structure is built on the set $P_i \subset P$ of all Voronoi neighbors of p_i (points of P whose Voronoi cell is adjacent to the Voronoi cell of p_i). For P_i , a new Voronoi diagram is built and this is also preprocessed for planar point location queries. This gives us the structure T_i . Note that every point p_i gives rise to a different point set P_i and a different structure T_i .

It is clear how a query with q is performed: first we find the the closest point and report it; assume it is some point p_j from P . Then we follow the pointer stored with the trapezoid of the cell of p_j , and query with q in T_j .

Give the storage requirements of this solution and prove it. Give the query time for finding the closest two points of a query point q and prove it as well. You may refer to results in the textbook without proving them, but you must specify clearly where you use a textbook result.

5. (1+1 points) Let R be the square with corners $(1, 1)$, $(-1, 1)$, $(-1, -1)$, and $(1, -1)$, and define $R(\alpha)$ to be the square obtained after rotating R by an angle $\alpha \in [0, \pi/2)$ about the origin.

(a.) Prove or disprove: There exists an $\alpha \in [0, \pi/2)$ such that $R(0) \oplus R(\alpha)$ has exactly eight vertices.

(b.) Prove or disprove: There exists an $\alpha \in [0, \pi/2)$ such that $R(0) \oplus R(\alpha)$ has less than eight vertices.

6. (2 points) Consider a set S of n disjoint line segments in general position in the plane, and a point p that is not on any of the line segments of S .

Give a plane sweep algorithm that, given S and p , computes all points at a given distance L from p that lie on a segment $s \in S$ and are visible from p . (Hint: use a rotating half-line emanating from p rather than a translating line to sweep the plane.) Your algorithm should run in $O(n \log n)$ time.

Be sure to discuss the sweep line status, the status structure, the event queue, the different types of events that occur, and how these events should be processed.