

Exam in Geometric Algorithm

Q1 — 20 points A *diamond* is defined as a square rotated in 45 degrees. Let S be a set of n points in the plane. Suggest a data structure that supports the following type in queries: For a given diamond D , report all the points in $S \cap D$. Answering a query should be done in time $O(\log^2 n + k)$, where k is the size of the output. The preprocessing time should be $O(n \log^2 n)$.

Answer: Rotate the coordination system by 45 degrees. Each diamond query now becomes orthogonal rectangle range query, after the obvious transformation. Use the orthogonal two-dimensional range query to support answering this query.

Q2 — 30 points Given a set S of n segment. Suggest a way to compute the visibility graph for S in time $O(n^2)$. Hint - use duality.

Q3 — 25 points

- Given a set S of n axis-parallel unit squares in the plane, and another point q . Describe the set T of all translations t such that $q + t$ is contained in any of the squares of S . (Hint — solve the question first in the case that S contains only one square).

Answer: Let U be the unit of the squares of S . Then $t + q \in U$ if and only $t = U - q$.

- What is the complexity of the union of n axis-parallel squares in the plane ? Answer in two ways:
 - Show that squares are pseudo-disks and use the appropriate claims from the textbook.
 - Show directly.

Answer: The complexity is linear, which would follows by showing that the set of unit squares are pseudo-disks. Let a, b be unit squares. If $a \setminus b$ consists of two connected component, then there must be an edge e of a for which its endpoints are in the different components of $a \setminus b$, but the length of e is 1 while the length of the portion of $e \setminus b$ is the height or width of b which is also 1. Contradiction. Thus $a \setminus b$ consists of at most one connected component.

To show the same bounds directly, let v be a vertex of ∂U , attached to an edge e of a . We follow e into b . We must reach a corner u of a before exiting b , since the length of e is 1. Thus we can charge u to v . Since the number of corners is $4n$, the number of vertices of ∂U is also $\leq 4n$.

Q4 — 25 points Let T_1 and T_2 be two Quad-Trees, defined on the same universe U . Each leaf in T_1 and in T_2 represent a block, which is either black or white. Let S_1 and S_2 be the shapes defined respectively as the union of black leaves in T_1 and T_2 , respectively. Assume that T_1 (resp. T_2) contains n_1 (resp. n_2) nodes. Show how in time $O(n_1 + n_2)$ you can compute the area of a the $S_1 \cap S_2$.

Answer: We traverse both tree simultaneously, maintaining the same left-to-right visit of the children in each tree. Let σ be a block corresponding to a node that we visit in one of the trees. Three cases might arise: (i) If σ is not a leaf in either trees, we continue the search in the children of σ . If σ is white in one trees, then no black point would be counted in $S_1 \cap S_2 \cap \sigma$, so we return to the parent of σ in both trees. If σ is black in one of the trees, then we visit all the leaves in the subtree of σ in the tree, and sum the area of the black leaves in this subtree. The total sum that we obtain is the area of $S_1 \cap S_2$.

Q5 — 20 points You are given a set S on n points in the plane, and two special points $s, t \in S$. The goal is to find a path from s to t , that might pass also through some other points in S . The cost of the path show be as small as possible. The cost of getting directly from some point $a \in S$ to a point $b \in S$, along a straight segment, equal to $dist^2(a, b)$. The cost of a path from s to t is the sum of the costs of the individual segments along the path. Describe an $O(n \log n)$ time algorithm for finding the path with minimal cost.

Answer: We compute the Delauney triangulation T of S . We can assume that the shortest path only uses edges from T . We can clearly assume that that path uses points of S as vertices. If it uses an edge ab which is not in T , then the circle whose diameter is e must contain another point c of S (otherwise it would be an Delauney edge). Then $|ac|^2 + |cb|^2 < |ab|^2$ so the path is not optimal.

Q6 — 20 points

- Show a set of n segments in the plane for which the algorithm for constructing BSP presented in the textbook would run long time, under unsuccessful choice of the order of the segment. Show what is the unfortunate order, and what is the maximal running time.

Answer: n horizontal and below them n vertical segments - and in the unfortunate order we use all the vertical one first. The ray from each of them split all the horizontal ones.

- How would you construct a BSP for a set of disjoint triangles in the plane? What is the size of this BSP?

Answer: Just use BSP for the edges of each triangle. The analysis is the same as for BSP for set of segments, since the number of triangles that a ray splits is proportional to the number of segments that it splits.

Q7 — 25 points Let S be a set of n points in the plane, no two with the same x or y coordinate. Suggest a data structure that supports the following operation: Given a query axis-parallel rectangle R in the plane, report the following points in $R \cap S$.

- The one with the maximal x coordinate
- The one with the minimal x coordinate
- The one with the maximal y coordinate
- The one with the minimal y coordinate.

The preprocessing time should be $O(n \log^2 n)$, and a query should be answered in time $O(\log^2 n)$.

Answer: Use a standard two-dimensional range tree, and which one tree T sorted by the x -coordination of the points, points to $O(n)$ trees, each storing a subsets of the points sorted by their y -coordination. To find the uppermost and lowest points, we perform we just use successor and predecessor operations in each of the trees T_i that we query, and take the maximum and minimum among the $O(\log n)$ values that we obtain in the $O(\log n)$ values of the trees T_i that we check.

To find the leftmost and rightmost points, we replace each T_i by a segment tree defined on the same set of points, and also sorted by the y -coordination. Each node μ in each one of these trees also maintain the leftmost and rightmost point in the set S_i . As shown in the class, we only need to check $O(\log n)$ nodes μ in each segment tree that we visit in the process of the search, and as before, the leftmost and rightmost point of the $O(\log n)$ nodes that we visit.

Other alternative for finding the leftmost and rightmost points, is to maintain another two dimensional range tree data structure where the primaty tree is sorted by the y -coordination, and the seconddary structure is sorted by x . Then the question is answer as in the first part of this answer.

Q8 — 28 points Let S be a set of n sites (points) in the plane. We define the *farthest-site Voronoi-Diagram* (FSVD) of S as a partition of the plane into regions, such that two points q_1, q_2 lie in the same region if the farthest site of S from q_1 and the the farthest site of S from q_2 is the same site.

- Assume that S contains only two sites. Describe the FSVD of S .
- It is known that the FSCD for S contains $O(n)$ vertices and edges. Assume you are given the FSVD of S . Give a linear time algorithm to find the smallest disk that contains all the site of S .

Answer: The voronoi diagram of two points contains only one edge, which is the bisector of the two points.

To find the smallest enclosing circle C , Observe first if x is any point and a is the farthest site from x , the a disk centered at x and of radius $|xa|$ must contains S . However, it might not be the smallest such disk.

observe that C must be determined by two points on its boundary, and then these two points forms its diameter, or three points a, b, c on its boundary. In the later case, observe that the center c of C must be of equal distance from a, b, c , and this distance is larger than its distance from any other point of S . Thus c must be a vertex of the FSVD of S . Since there are only $O(n)$ such vertices, we check the distance from each to one of the sites defining it, and take the one for which the minimum is obtained, since this is the smallest enclosing disk.

In the former case, that C is defined by two sites, its center c must lie on an edge of the diagram. Similar check to the one above reveals where the optimal point must lie on this edge. Similarly we checked all edges to find the optimal one.