CMSC 754 Quiz 2

This quiz is closed-book and closed-notes, but you may use one sheet of notes, front and back. The total point value is 50 points.

Unless otherwise stated, you may assume that inputs are in *general position*. If you are asked for an $O(T(n))$ time algorithm, you may give a *randomized* algorithm with *expected* time $O(T(n))$. If you are asked to give an algorithm, also explain how it works and derive its running time. (Unless otherwise stated, formal proofs of correctness are not required.) You may make use of any results presented in class and any well-known facts from algorithms or data structures.

Problem 1: (10 points) Short-answer questions. Unless requested, explanations are not needed.

(a) (2 points) The following recurrence appeared in one of our algorithm analyses. Which one?

$$T_d(n) \;=\; T_d(n-1) + d + p_n(dn + T_{d-1}(n-1))$$

(i) Linear Programming (LP)
(ii) Trapezoidal Map
(iii) Point-Location
(iv) Voronoi Diagram (Fortune's algorithm)

(b) (2 points) Assuming $p_n = d/n$, what did our analysis reveal the solution of the above recurrence to be? Select one:

(i) $O(dn)$ (ii) $O(n \log^d n)$ (iii) $O(d!n)$ (iv) $O(n^{\lfloor d/2 \rfloor})$

(c) (2 points) In Fortune's algorithm for Voronoi diagrams, you have just swept through i (out of n) sites. What is the *maximum* number of arcs that a *single site* can contribute to the beach line? Select one:

(i) 1 (ii) $O(1)$ (iii) $O(i)$ (iv) $O(n-i)$ (v) $O(n)$

(d) (2 points) Assuming segments are inserted in *random order*, what is the *expected* number of *new trapezoids* created when the ith (out of n) segment is inserted? Select one:

(i) 1 (ii) $O(1)$ (iii) $O(\log i)$ (iv) $O(\log n)$ (v) $O(i)$ (vi) $O(n)$

(e) (2 points) Assuming segments are inserted in *arbitrary order*, what is the *worst-case* number of *new trapezoids* created when the ith (out of n) segment is inserted? Select one:

(i) 1 (ii) $O(1)$ (iii) $O(\log i)$ (iv) $O(\log n)$ (v) $O(i)$ (vi) $O(n)$

Problem 2. (10 points) You are given a set of n lines in $\mathbb{R}^2$, where the ith line is $\ell_i : y = a_i x + b_i$. Present an efficient algorithm that computes the largest square containing the origin, such that no line of the set intersects the square's interior (see Fig. 1(a)). You may assume that no line passes through the origin. (**Hint:** $O(n)$ time is achievable.)

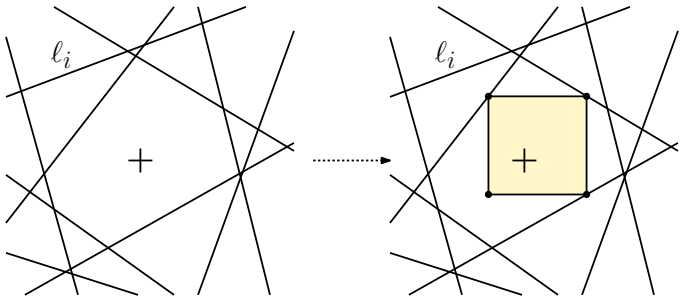


Figure 1: Largest square.

Problem 3. (15 points) You are given a set of n circular disks in the plane, each of radius 1, centered at a given set of points $P = \{p_1, \ldots, p_n\}$ (see Fig. 2(a)). Any point where the boundaries of two disks intersect is called a *vertex*. A vertex is *exposed* if it lies on the boundary of the union of the disks, and otherwise it is *hidden* (see Fig. 2(b)).

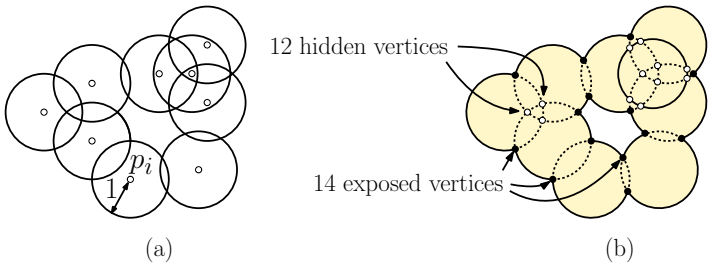


Figure 2: Hidden and exposed vertices in a set of unit disks.

(a) (4 points) Which of the following most accurately reflects the **maximum** possible number of **exposed** vertices? Select one.

(i) $O(n)$ (ii) $O(n \log n)$ (iii) $O(n^2)$ (iv) $O(2^n)$

(b) (4 points) Which of the following most accurately reflects the **maximum** possible number of **hidden** vertices? Select one.

(i) $O(n)$ (ii) $O(n \log n)$ (iii) $O(n^2)$ (iv) $O(2^n)$

(c) (7 points) Present an efficient algorithm that, given the center points $P = \{p_1, \ldots, p_n\}$, returns the **number of exposed vertices**. Briefly justify its correctness and derive its running time. (**Hint:** $O(n \log n)$ time is possible.)

Problem 4. (15 points) You are given a set of n vertical line segments all in the positive quadrant of the x, y-plane, where the ith segment is specified by giving its center point $p_i = (p_{i,x}, p_{i,y})$ and its height h_i (see Fig. 3(a)).

(a) (10 points) Explain how to construct a data structure so that, given any query ray emanating from the y-axis, it is possible to report the first (leftmost) segment, if any,

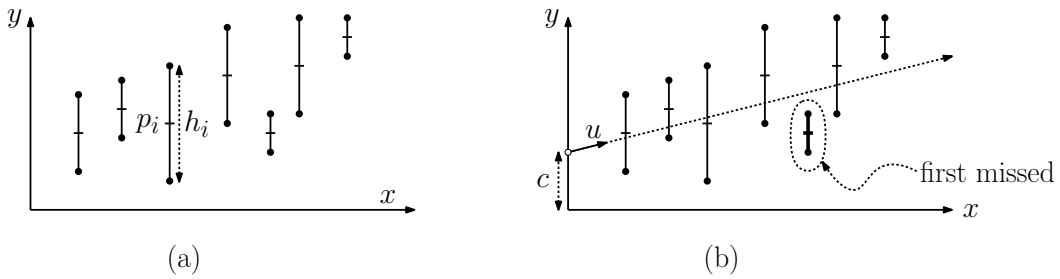


Figure 3: Report the first miss.

that is *missed* by a given ray (see Fig. 3(b)). The ray is given by its y-intercept, denoted c, and a directional vector $u = (u_x, u_y)$, where $u_x > 0$. Your data structure should have storage $O(n)$ and query time $O(\log n)$.

(b) (5 points) Suppose that instead, you were asked to present a data structure that determines the first (leftmost) segment that the ray *hits*, if any, in $O(\log n)$ time (see Fig. 4). Show that it is **not possible** to do this via point location with $O(n)$ space in the worst case. (**Hint:** Show that there exists a configuration of n segments such that the solution space has complexity that is super-linear in n.)

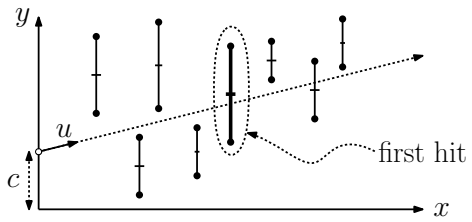


Figure 4: Report the first hit.