CMSC 754 Quiz 1

This quiz is closed-book and closed-notes, but you may use one sheet of notes, front and back. You may use any algorithms or results given in class. The total point value is 50 points. Good luck!

Problem 1: (18 points) Short-answer questions. Unless requested, explanations are not needed.

(a) (2 points) Consider the execution of the plane-sweep algorithm from class for decomposing a polygon into x-monotone pieces shown in Fig. 1(a). Which of the given vertices (a, b, c, or d) is helper(e)?

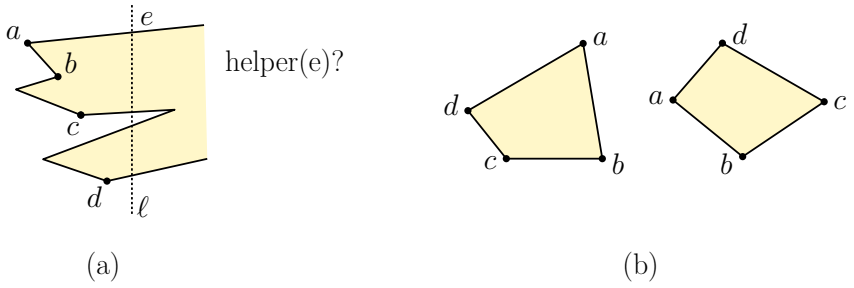


Figure 1: Short-answer questions.

(b) (8 points) You are given a sequence of four points in the plane, $\langle a, b, c, d\rangle$. Using orientation tests alone, present a test that these points are the vertices of a convex quadrilateral, given in cyclic order, either clockwise or counterclockwise (see Fig. 1(b)).

(c) (8 points) Consider the dual transformation mapping the point $p = (a, b)$ to the line $p^* : y = ax - b$. For each of the options below, .

- If p_i and p_j are on the **upper** hull of P with p_i to the **left** of p_j, then p_i^* and p_j^* are on the (upper) / (lower) envelope of P^* with p_i^* to the (left) / (right) of p_j^*.
- If p_i and p_j are on the **lower** hull of P with p_i to the **left** of p_j, then p_i^* and p_j^* are on the (upper) / (lower) envelope of P^* with p_i^* to the (left) / (right) of p_j^*.

Problem 2. (8 points) Consider a set $P = \{p_1, \ldots, p_n\}$ of points in the plane, where $p_i = (x_i, y_i)$. A *Pareto set* for P, denoted Pareto(P), is a subset of points p_i such that there is no $p_j \in P$ ($j \neq i$) such that $x_j \geq x_i$ and $y_j \geq y_i$ (see Fig. 2(a)).

Present a *divide-and-conquer* algorithm for computing Pareto(P) in $O(n \log n)$ time. Your algorithm should first partition the set into two roughly equal-sized subsets P_1 and P_2 (arbitrarily, without sorting), recursively compute the Pareto sets of each subset, and then merge the two Pareto sets in $O(n)$ time (see Fig. 2(b)). The output should be presented as a sequence of indices of the points along the Pareto set, sorted from left to right. Briefly justify your algorithm's correctness and derive its running time.

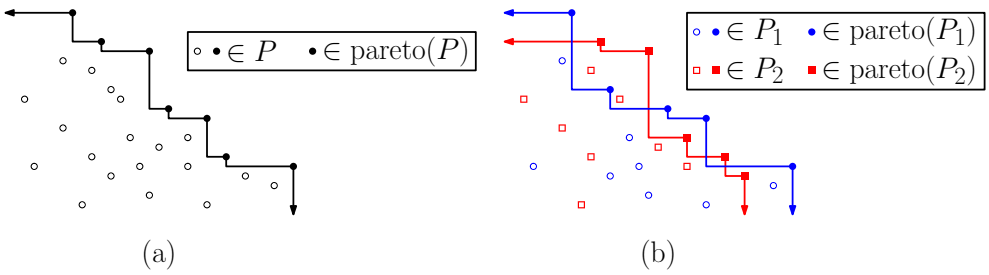


Figure 2: Divide and conquer algorithm for Pareto sets.

Problem 3. (8 points) You are given a square in the plane (e.g., the unit square $[0,1] \times [0,1]$). You are also given two sets of line segments, R and B (for *red* and *blue*) of sizes n and m, respectively, each of which has endpoints on the boundary of the square. All the segments have their left endpoint on the square's left side. The red segments are pairwise disjoint, and all have positive slope. The blue segments are pairwise disjoint, and all have negative slopes (see Fig. 3).

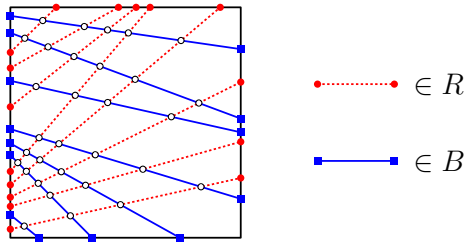


Figure 3: Counting red-blue segment intersections.

Present an efficient algorithm that *counts* all the red-blue intersections. Justify your algorithm's correctness and derive its running time. For full credit, your algorithm should run in $O((n+m)\log(n+m))$ time, irrespective of the number of intersections (which could be as large as nm).

Problem 4. (16 points) You are given a simple polygon P in the plane which may contain disjoint polygonal holes. You are also given two points s and t lying within P's interior, where s's x-coordinate is smaller than t's x-coordinate. Let n denote the total number of vertices, including those of the holes. This problem considers x-monotone paths from s to t that lie entirely within P's interior.

Provide an answer for *one* of the following two problems, and indicate which problem you are solving. (**Hint:** Both can be solved by plane sweep in $O(n \log n)$ time.)

Simple version: (For half credit) Present an efficient algorithm which, given P, s and t, reports whether there *exists* an x-monotone path from s to t. (It does not need to output the actual path.) Briefly justify your algorithm's correctness and derive its running time.

Full version: (For full credit) For each hole whose x-extents lie between s and t, any x-monotone path from s to t either passes above the hole or below it. We say that two

x-monotone paths are *equivalent* if they pass above the same set of holes and below the same set of holes. (In Fig. 4(a) we show three equivalent paths, and in Fig. 4(b) we show four paths from different equivalence classes.)

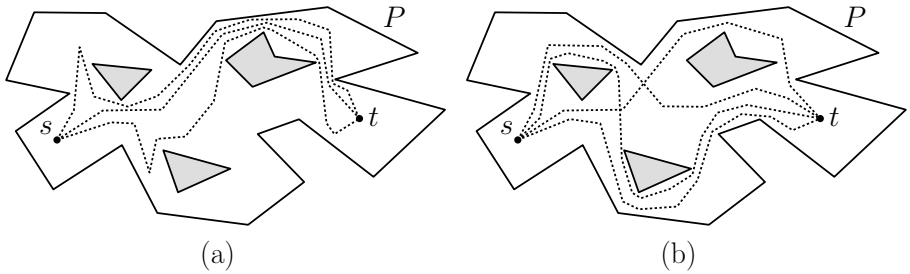


Figure 4: Equivalence classes for x-monotone paths.

While there are an infinite number of distinct x-monotone paths from s to t, there are only a finite number of equivalence classes. Present an efficient algorithm which, given P, s and t, returns a *count* of the total number of equivalence classes of x-monotone paths from s to t. Briefly justify your algorithm's correctness and derive its running time.