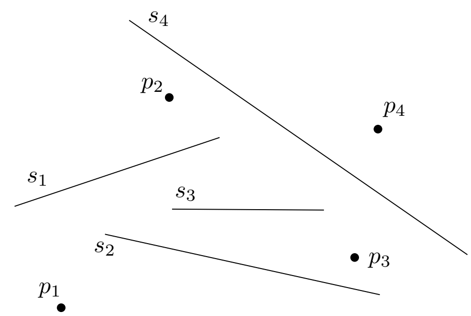
CSCI5010 Exercise List 9

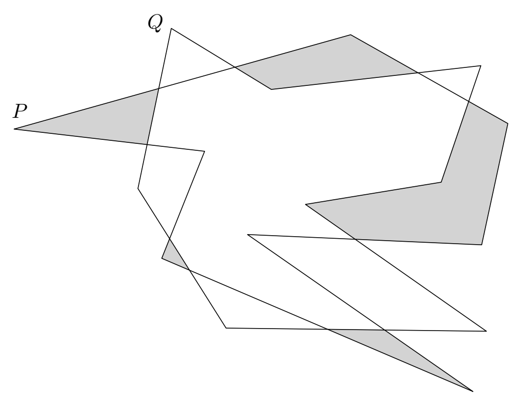
Problem 1. Give a *deterministic* algorithm to compute the trapezoidal map of n non-intersecting segments in $\mathbb{R}^2$ using $O(n \log n)$ worst case time (hint: planesweep).

Problem 2 (Batched Point Location). Let S be a set of n non-intersecting segments, and P a set of m points, all in $\mathbb{R}^2$. Give an algorithm to compute, for each point $p \in P$, the lowest segment (if any) in S that is hit by the upward ray emanating from p. Your algorithm must finish in $O(n \log n + m \log m)$ time.



In the above example (where $n = m = 4$), your algorithm should report s_1 for p_1, s_4 for p_2 and p_3, and nothing for p_4.

Problem 3. Let P and Q be two arbitrary (possibly concave) polygons in $\mathbb{R}^2$. The vertices of each polygon are given in clockwise order in an array. Describe an algorithm to compute the area (e.g., where the word "area" is as in "the area of a square with side length 2 is 4") of the region that is inside P, but outside Q. Your algorithm should finish in $O(n \log n + k \log k)$ time, where n is the total number of vertices of P and Q, and k is the number of intersection points between the edges of P and Q.



In the above example, $n = 16$ (P and Q are octagons) and $k = 10$. Your algorithm should calculate the area of the shaded region.

(Hint: first compute all the intersection points, and then, compute a trapezoidal map. The challenge is to determine whether each trapezoid is inside the designated region; use planesweep to tackle the challenge.)