Computational Geometry
Homework 1

Several problems in this homework are adapted from the textbook *Computational Geometry: Algorithms and Applications* by de Berg et al., but I have restated the problems for convenience.

1. Show that the point $r = (r_x, r_y)$ lies to the left of the directed line from $p = (p_x, p_y)$ to $q = (q_x, q_y)$ if and only if the expression

   $$(q_x - p_x)(r_y - p_y) - (r_x - p_x)(q_y - p_y)$$

   is positive. Optional: Also verify that checking the latter is equivalent to computing the sign of the determinant

   $$\begin{vmatrix} 1 & p_x & p_y \\ 1 & q_x & q_y \\ 1 & r_x & r_y \end{vmatrix}$$

2. Let S be a set of n disjoint line segments whose upper endpoints lie on the line $y = 1$ and whose lower endpoints lie on the line $y = 0$. These segments partition the horizontal strip $[-\infty, +\infty] \times [0, 1]$ into $n+1$ regions. Give an $O(n \log n)$ time algorithm that takes as input S and creates a data structure, so that given any query point q in the strip, the region containing q can be determined in $O(\log n)$ time. Describe the query algorithm in some detail.

   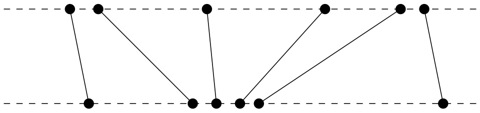

   Figure 1: Problem 2

   For giving the answer to a query, you can "label" a region in any way that you deem reasonable. Also make a reasonable choice for what to output for a query point that lies on some segment.

3. Let S be a set of n disjoint line segments in the plane, and let p be a point not on any of the line segments of S. We wish to determine all line segments of S that p can see, that is, all line segments of S that contain some point q so that the open segment $\overline{pq}$ doesn't intersect any line segment of S. See accompanying figure for an illustration. Give an $O(n \log n)$ time algorithm for this problem that uses a rotating half-line with its endpoint at p. (4 points)

   This problem can be solved via a modification of the plane-sweep approach. Here, you should describe the sweep-line status (or its equivalent), the event points, the data

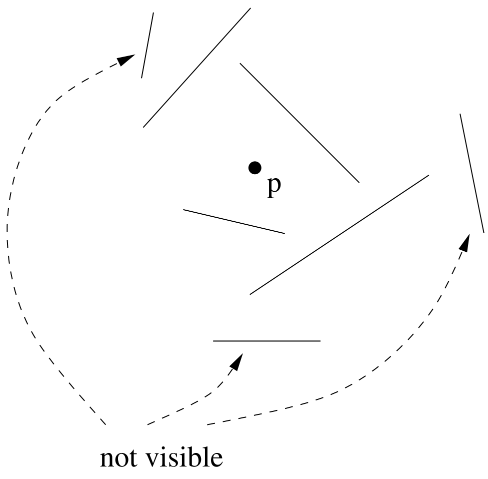

Figure 2: Problem 3

structures used for maintaining these, and the actions taken at each type of event point. You should also outline how the desired running time bound follows. I am not looking for great detail, but for the key ideas.

4. Suppose that we have a subroutine ConvexHull available for computing the convex hull of a set of points in the plane. Its output is a list of convex hull vertices, sorted in clockwise order. Now let $S = \{x_1, x_2, \ldots, x_n\}$ be a set of n integers. Show that S can be sorted in $O(n)$ time plus the time needed for one call to ConvexHull.

5. Show that for a connected planar graph with $v \geq 1$ vertices, e edges, and f faces, we must have $v - e + f = 2$. (Hint: Start with the trivial graph with only one vertex, build up the graph while maintaining connectivity, and ask what happens to $v - e + f$.)

The homework is due via ICON by 11:59 pm on Thursday, February 6. I would prefer if you type-set your submissions, but feel free to include hand-drawn figures.

Please pay attention to the policy on collaboration outlined in the syllabus. In particular, you will get the most out of home work if either you work on your own or collaborate with a fellow student or two in a reasonable way. I will evaluate the originality of your writing while reviewing your work. Feel free to stop by during my walk-in hours if you need help.