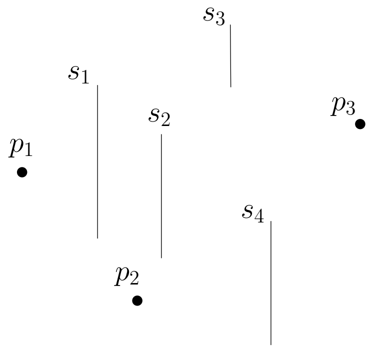
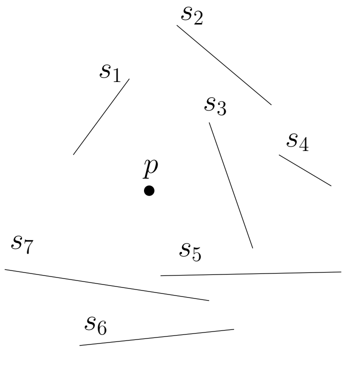
# Exercise List 3

**Problem 1 (Range Max).** Let $S$ be a set of $n$ real numbers. Each number $v \in S$ is associated with a real valued *weight*. Given a range $[x, y]$, a query returns an element in $S \cap [x, y]$ with the maximum weight. For example, if $S = \{(1, 15), (3, 7), (7, 12), (10, 9)\}$, where each pair has the form $(v, weight(v))$. Then, a query with range $[2, 15]$ returns $(7, 12)$. Design a data structure to answer such queries in $O(\log n)$ time. Your structure should also support insertions and deletions in $O(\log n)$ time.

**Problem 2 (Batched Line Dragging).** Let $S$ be a set of $n$ vertical line segments in $\mathbb{R}^2$ (i.e., each segment has the form $x \times [y_1, y_2]$). Also, let $P$ be a set of $m$ points in $\mathbb{R}^2$. For each segment $s \in S$, we want to output a pair $(s, p)$ where $p$ is the first point in $P$ that is hit by $s$ if $s$ moves left; if $p$ does not exist, output $(s, nil)$. Describe an algorithm to do so in $O(n \log n + m \log m)$ time. For example, in the following figure, you should output $\{(s_1, p_1), (s_2, p_1), (s_3, nil), (s_4, p_2)\}$. You may assume that $P$ is in general position (i.e., no two points have the same x-coordinate or y-coordinate).



**Problem 3 (Rotating Sweep; Exercise 2.14 from textbook).** 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 in $S$. We want to determine all line segments of $S$ that $p$ can see, that is, all line segments of $S$ that contain some point $q$ so the segment $pq$ does not intersect any segment in $S$ (except at $q$, of course). Give an $O(n \log n)$ time algorithm to solve the problem. For example, in the following figure, you should output all segments but $s_4$ and $s_6$.


$s_2$
$s_1$
$s_3$
$s_4$
$p$
$s_7$
$s_5$
$s_6$