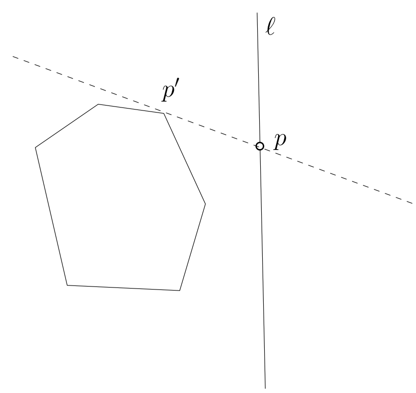
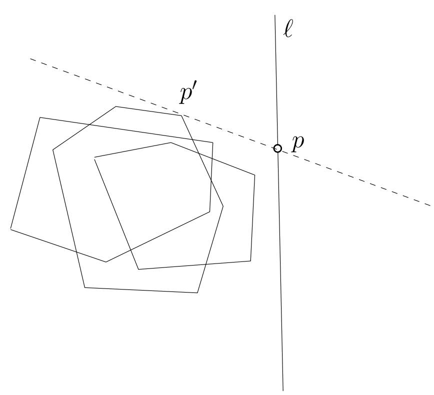
Exercises

Problem 1 (General Binary Search). Let A be an array of n real values. A has the property that if we start from some position and then look at these values in a cyclic manner, we see a pattern where the values initially increase monotonically and then decrease monotonically. For example, A can be $(10, 20, 30, 25, 15, 0, 5)$; namely, if we inspect the values in this order: 0, 5, 10, 20, 30, 25, 15, then we observe the pattern mentioned earlier. On the other hand, A does not have the property if its values are $(5, 20, 30, 25, 15, 0, 10)$. Design an algorithm to find the maximum value in A in $O(\log n)$ time (note that you do *not* know where is the "starting position" mentioned earlier).

Problem 2 (Gift Wrap). Let P be a convex polygon of n vertices which have been stored in an array in the counterclockwise order. Let ℓ be a line in the plane such that the entire P falls on the left side of ℓ. Now, fix a point p on ℓ. We want to turn ℓ counterclockwise with p as the pivot, and stop as soon as ℓ hits the first vertex of P (e.g., in the figure below, the answer is p'). Design an algorithm to find in $O(\log n)$ time the first vertex hit.



Problem 3 (Gift Wrap Again). Let $P_1, ..., P_m$ be m arbitrary convex polygons, each of which has no more than k points. The vertices of each polygon have been stored in an array in the counterclockwise order. Let ℓ be a line in the plane such that all the $P_1, ..., P_m$ fall on the left side of ℓ. Now, fix a point p on ℓ. We want to turn ℓ counterclockwise with p as the pivot, and stop as soon as ℓ hits the first vertex of any polygon (e.g., in the figure below, the answer is p'). Design an algorithm to find in $O(m \log k)$ time the first vertex hit.



Problem 4 (Output-Sensitive Convex Hull). Let S be a set of n points in $\mathbb{R}^2$. You are given an integer $\hat{k}$ that is guaranteed to be larger than or equal to the number of vertices on the convex

hull of S. Give an algorithm that computes the convex hull in $O(n \log \hat{k})$ time. (Hint: arbitrarily divide S into groups of size $\hat{k}$.)