1. [10 marks] The **Smallest Triangle Problem** is to find, given a set $S$ of $n$ points in the plane, three points of $S$ that determine the smallest area triangle. This is a generalization of the problem of testing if three points are collinear (because collinear points give a triangle of area 0). In this question you will use duality and arrangements to solve the Smallest Triangle Problem in $O(n^2)$ time. Assume that no two points of $S$ have the same $x$-coordinate.

(a) First suppose that two points $a$ and $b$ in $S$ are fixed. The goal is to find the point $c \in S$ to minimize the area of triangle $abc$. (Yes, the problem can then trivially be solved in linear time, but we’ll still look at the dual.) Let $\ell$ be the line through $a$ and $b$. For any point $p \in S$, let $\ell_p$ be the line through $p$ parallel to $\ell$.
Prove that $c$ is the point such that $\ell_c$ is closest to $\ell$. (Don’t belabour this, it’s high-school geometry.)
Express this in terms of the dual, with lines $a^*, b^*, c^*$ and points $\ell^*, \ell_c^*$. (Remember what happens to parallel lines when you dualize. Draw a figure.)
Describe how to find $c^*$ in the dual arrangement.

(b) Give an $O(n^2)$ time algorithm to solve the Smallest Triangle Problem by constructing the dual arrangement. Give a high-level description of your algorithm, not detailed pseudo-code. **Hint:** You will need to revisit the algorithm that constructs the arrangement so you can collect the information that was useful in part (a).

2. [10 marks] Design a polynomial time algorithm to find a path from point $s$ to point $t$ among disjoint disc obstacles in the plane. Do not invest too much energy in the best run time, but do be sure to justify correctness. Give a high-level description of your algorithm, not detailed pseudo-code. You may assume some geometric primitives for pairs of discs without giving details—but be sure to say what geometric primitives you assume.