Due date: February 17, 2022

Problem 1: [15 pts] Let $\triangle$ be a triangle in $\mathbb{R}^2$. For a point $x \in \mathbb{R}^2$, $\triangle + x$ is a *translate* of $\triangle$ (if a, b, c are vertices of $\triangle$, then the vertices of $\triangle + x$ are $a + x, b + x, c + x$.) Let S be a set of n translates of Δ. Let $U(S) \subseteq \mathbb{R}^2$ denote the union of triangles in S, i.e., the set of points that lie in at least one triangle of S. Each connected component of $U(S)$ is a polygon (with possible holes). The total number of vertices on $U(S)$ is known to be $O(n)$. Describe an $O(n \log^2 n)$-time algorithm to compute the boundary of the union of $\mathcal{S}$. (You can assume that once you have all the edges of a planar subdivision Π, one can compute a suitable representation of Π in linear time.) (**Hint:** *Describe a divide-and-conquer algorithm and use sweep-line at the merge step.*)

Problem 2: [10 pts] Let S be a set of (possibly intersecting) m line segments in $\mathbb{R}^2$, and let L be a set of n lines in $\mathbb{R}^2$. Describe an $O((m^2 + n) \log n)$ algorithm to count the number of intersection points between L and S. (**Hint:** *Use duality and line sweep.*)

Problem 3: [10 pts] Let C_1 and C_2 be two disjoint convex polygons, each with n vertices. Assume that the vertices of each polygon are stored in counterclockwise direction in a height-balanced tree (e.g. red-black tree). Describe an $O(\log n)$-time algorithm to compute their outer tangents. (**Hint:** *How does one determine which paths one should follow in the two trees to compute the upper tangent?*)

Problem 4: [15 pts] Let $P \in \mathbb{R}^2$ be a set of n points. The *width* of P is the minimum width of a strip (region bounded by two parallel lines) that contains P.

(i) Show that the width of P can be computed in $O(n \log n)$ time.

(ii) Extend (i) to compute the smallest area rectangle that contains P in $O(n \log n)$ time.