Exercises for CSCI5010

Prepared by Yufei Tao

Problem 1. Let P be a set of n points in $\mathbb{R}^2$. Describe how to compute in $O(n)$ time a triangle that includes all the points of P in the interior.

Hint: First compute an axis-parallel rectangle that cover all the points of P.

Problem 2*. Let $G = (V, E)$ be a connected regular straight-line planar graph (SLPG) with $n = |E|$ segments. Explain how to compute in $O(n)$ time a triangulated SLPG $G' = (V', E')$ such that

- V' includes all the vertices of V, plus three dummy vertices that determine a triangle covering all the points of V in interior;
- $E \subseteq E'$;
- Every face of G' is covered by a face of G.

Hint: Identify the letmost and rightmost points in V. Then, find a triangle Δ that includes all the points of V in the interior. The remaining obstacle is to triangulate the area "between" G and the triangle's boundary. This obstacle can be tackled by adding two segments, the first of which connects the leftmost point of V to a vertex of Δ, while the other connects the rightmost point of V to another vertex of Δ. Now, recall that an x-monotone polygon can be triangulated in linear time.

Problem 3. Let G be a connected regular SLPG with n segments. Describe how to build the point-location structure we discussed in $O(n \log n)$ time.

Problem 4. Prove: If a triangulated SLPG has n vertices and m edges, it must hold that $m = 3n - 6$.

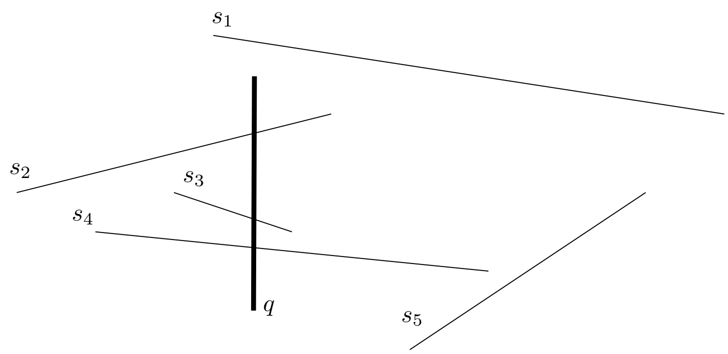
Hint: Recall that the outer face of a triangulated SLPG is a triangle. Apply induction.

Problem 5 (Reading Exercise). Prove: The trapezoidal map defined by n non-intersecting line segments in $\mathbb{R}^2$ has complexity $O(n)$.

Hint: Page 56 of Prof. Mount's notes.

Problem 6. Describe an algorithm to build the trapezoidal map from n non-intersecting line segments in $\mathbb{R}^2$ using $O(n \log n)$ time.

Problem 7*. Let S be a set of n non-intersecting line segments in $\mathbb{R}^2$. Given a vertical segment q, a query retrieves all the segments of S intersecting q. Design a data structure of $O(n)$ space that answers a query in $O(\log n \cdot (1 + k))$ time, where k is the number of segments reported. In the following example where $S = \{s_1, s_2, ..., s_5\}$, the query q retrieves $k = 3$ segments.


s_1
s_2
s_3
s_4
q
s_5