Exercises for CSCI5010

Prepared by Yufei Tao

Problem 1 (Top-1 Search). Let P be a set of n points in $\mathbb{R}^2$. Let x_p (resp., y_p) denote the x- (resp., y-) coordinate of p. Define a *preference function* to be a function $f : \mathbb{R}^2 \to \mathbb{R}$ of the form: $f(p) = c_1 \cdot x_p + c_2 \cdot y_p$, where c_1 and c_2 are real-valued constants. Given a preference function f, a *top-1 query* returns a point $p \in P$ that maximizes $f(p)$ among all the points in P.

Design a structure of $O(n)$ space that answers a query in $O(\log n)$ time. Describe how to construct the structure in $O(n \log n)$ time.

Problem 2 (Merging Convex Hulls). Let P_1 and P_2 be two sets of points. Given the convex hulls of P_1 and P_2, describe an algorithm to compute the convex hull of $P_1 \cup P_2$ in $O(n)$ time, where $n = |P_1| + |P_2|$.

Remark: This implies an $O(n \log n)$ time divide-and-conquer algorithm for computing the convex hull of n points.

Problem 3. Prove: every polygon (which may be concave) with $n \geq 4$ vertices has at least one diagonal.

Problem 4. Consider the following algorithm for triangulating a polygon G:

```
1. add diagonals to break G into non-overlapping polygons G_1, G_2, ..., G_t without split vertices
2. for i = 1 to t do
3.     add diagonals to break G_i into non-overlapping polygons without merge vertices
4. for every polygon G' obtained at Line 3 do
5.     triangulate G' using an x-monotone algorithm
```

Prove: the above algorithm runs in $O(n \log n)$ time where n is the number of vertices in G.

Problem 5* (Point in Polygon). Let G be a convex polygon of n vertices, which are given to you in clockwise order. Given an arbitrary point $q \in \mathbb{R}^2$, describe an algorithm to decide whether q is inside or outside G in $O(\log n)$ time.

Problem 6* (Textbook Exercise 3.11). Given a polygon G of n vertices, decide in $O(n)$ time whether G can be made x-monotone by rotating the coordinate system at the origin.

Problem 7* (Reading Exercise). Let G be a polygon with n vertices. Two points p and q in the polygon are *visible* to each other if the the segment $\overline{pq}$ is fully contained by the polygon. Given a set S of vertices of G, we say that S *guards* G if every point inside G is visible to at least one vertex in G. Give an $O(n \log n)$ time algorithm to find a set S of size at most $n/3$ to guard G.

Hint: Read Section 3.1 of the textbook.