CS 598TMC, Fall 2021
Timothy Chan

Homework 2 (due Oct 20 Wednesday 5pm)

Instructions: see previous homework.

1. [*30 pts*] Given a convex polygon P with n vertices with integer coordinates in $[U]^2$, we want to build a data structure so that for any query line ℓ, we can quickly compute the intersection of P and ℓ (there are either zero or two intersection points).

   Show that this problem can be solved with $O(n)$ space and $O(\sqrt{\log U})$ query time. (Note that $O(\log n)$ time is not difficult, by binary search.)

   Hint: focus on computing the left intersection point. Use duality and a result from class.

   Research questions (both open, I think): is $O(\log^{1/2-\varepsilon} U)$ time possible with $O(n)$ space? what about the 3D case, where P is a convex polyhedron?

2. [*30 pts*] We are given a set P of points and a set H of halfplanes in 2D. The *depth* of a point p is defined as the number of halfplanes of H containing p. We want to find the maximum depth $D(P, H)$ over all points in P.

   Present a dynamic data structure that maintains the maximum depth $D(P, H)$ under insertions and deletions of halfplanes in H. Each insertion/deletion should take sublinear time.

   Hint: use a modified partition tree.

3. [*40 pts*]

   (a) [*30 pts*] Given n (possibly intersecting) line segments in 2D, we want to build a data structure so that for any query point q, we can count the number of line segments below q.

   Describe a solution with $O(n)$ or $O(n \log n)$ space and sublinear query time.

   Hint: combine a segment tree with partition tree.

   (b) [*10 pts*] Given n (possibly intersecting) triangles in 2D, we want to build a data structure so that for any query point q, we can count the number of triangles containing q.

   Hint: this should be easy, using (a) as a black box.