2.1 High-Incidence Line (8.14) [5]

Given a set of \( n \) points in the plane, find in \( O(n^2) \) time a line that is incident on the maximum number of points.

2.2 Minimum Area Triangle [10]

Use duality to compute the triangle of minimum area formed by a collection of \( n \) points in \( O(n^2) \) time.

2.3 Structure of Bad Triangles [15]

When we insert a point \( p \) into a Delaunay triangulation \( DT \), some of the triangles are now bad, in that their circumcircle contains \( p \). Consider the dual graph \( DT^* \) of \( DT \), formed by creating one node for each triangle of \( DT \), and connecting two nodes if the corresponding triangles share an edge. A node in \( DT^* \) is bad if its dual triangle is bad.

![Figure 2.1: A Delaunay triangulation and its dual](image)

1. Prove that in \( DT^* \), the set of bad nodes forms a connected subgraph.
2. All nodes in $\text{DT}^*$ have degree at most 3 (why?). Show how to output the above set of bad nodes in time linear in the size of the set, if you are given the triangle containing $p$.

2.4 Delaunay Triangulation of a Convex Polygon [20]

Consider the following algorithm to compute the DT of a convex polygon $S$, whose points are given in cyclic order.

1. If $S$ has at most three points, return $S$.
2. If not, pick a random point $q$ and compute recursively $\text{DT}(S' = S - \{q\})$.
3. Let $p, r$ be the two points on either side of $q$ in the order. Add the triangle $\Delta pqr$ to $\text{DT}(S')$.
4. Mark all bad triangles.
5. Remove all edges that touch only bad triangles, and retriangulate the face by adding diagonals from all points to $q$.

Use backwards analysis to show that this algorithm runs in expected linear time. You will need your solution to Problem 2.3. Another useful fact you can use without proof is that the Delaunay triangulation of a convex polygon on $n$ vertices is an outerplanar graph that has at most $2n - 3$ edges. (An outerplanar graph is a planar graph where all vertices lie on a single cycle).

Note that if we unwrap the recursion, this is a randomized incremental construction. This algorithm was designed by Paul Chew in the late 1980s.