CS 498 TC ✧ Computational Geometry ✧ Spring 2021

Homework 3

Due Monday, March 29, 2020 at 8pm

---

1. Suppose you are given a set of n circular disks in the plane, all with the same radius. Describe and analyze an efficient algorithm to determine whether the union of these disks is connected. For full credit your algorithm should run in $O(n \log n)$ time.

2. Suppose you are teaching a computational geometry class, and you have asked your students to implement an algorithm to construct Voronoi diagrams. You're allowing the students to use their own favorite programming language and programming environment, which means you can't compile the code yourself. Thus, instead of submitting the code itself, you asked your students to submit several Voronoi diagrams computed by their code.

   After the submission deadline, you realize that you forgot to ask the students to submit the *sites* that defined each of their Voronoi diagrams! In frustration, you decide to give your students full credit if every diagram they submit is the Voronoi diagram of *some* point set. But how do you tell?

   (a) Describe and analyze an algorithm to determine, given a planar straight-line graph G, whether G is a Voronoi diagram. Your algorithm should either return a finite point set P such that G is the Voronoi diagram of P, or report correctly that no such point set exists. *[Hint: Consider the case of four sites.]*

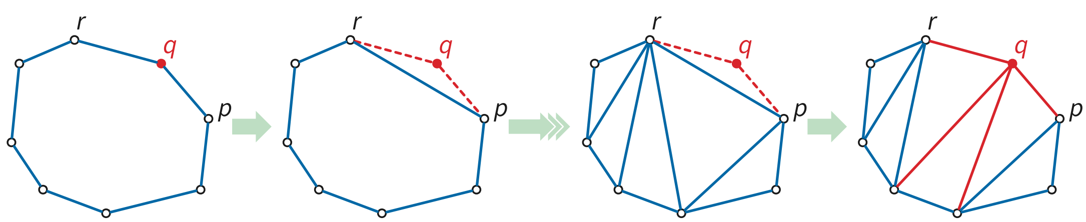
   (b) Generalize your algorithm from part (a) to *weighted* Voronoi diagrams (also known as *power diagrams*). Your new algorithm should either return a finite *weighted* point set P such that G is the weighted Voronoi diagram of P, or report correctly that no such weighted point set exists. *[Hint: Consider the case of five sites.]*

   *(c) **Extra credit:** Now consider the dual versions of parts (a) and (b). We can verify that a given triangulation is Delaunay if every pair of incident faces passes the empty circle test. We can similarly determine whether a given triangulation is a *weighted* Delaunay triangulation (that is, the dual of a power diagram) *if we are also given the weights of its vertices*. Lift each vertex (x, y) with weight w to the point $(x, y, x^2 + y^2 - w^2)$, and then check whether the resulting lifted triangulation is convex. But what if we are *not* given the weights w?

   Describe an efficient algorithm to determine, given a triangulation T of a set of n points *with integer coordinates*, whether T is a weighted Delaunay triangulation. Your algorithm should either report an appropriate weight assignment, or report correctly that no such assignment exists.

3. Let P be a convex polygon in the plane with n vertices, represented as a circular doubly-linked list of vertices in counterclockwise order. The following randomized incremental algorithm constructs the Delaunay triangulation of the vertices of P.

```
ConvexDelaunay(P):
  if P is a triangle
      return P
  else
      q ← random vertex of P
      p ← pred(q);  r ← succ(q)
      P′ ← P − pq − qr + pr
      T′ ← ConvexDelaunay(P′)
      T ← T′ + pq + qr
      repair T by flipping non-Delaunay edges opposite q
      return T
```



(a) Prove that the expected number of flips performed by this algorithm is $O(n)$.

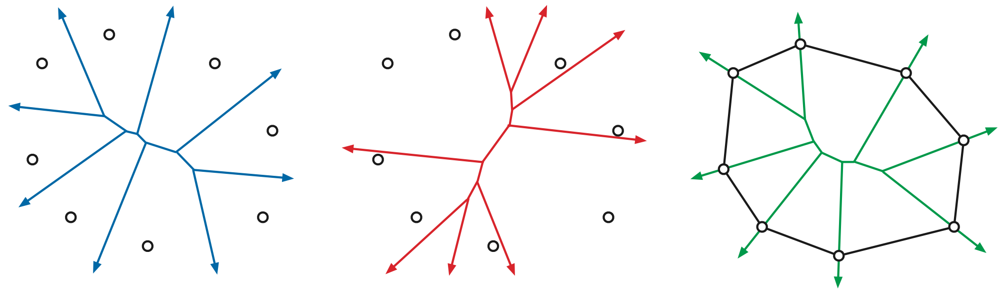
(b) We already know from Lawson's algorithm that we can perform the necessary flips in $O(1)$ time each. But how do we choose the random vertex q in $O(1)$ time?

4. Describe and analyze algorithms for each of the following natural variants of problem 3. The input to each problem is a convex polygon P, represented as a doubly-linked list of n vertices in counterclockwise order. For full credit, each of your algorithms should run in $O(n)$ expected time. Describe only the necessary changes to the algorithm from problem 3 and its analysis.

(a) Describe and analyze an algorithm to construct the *Voronoi diagram* of the vertices of P. *[Hint: This should be* ***very*** *easy!]*

(b) Describe and analyze an algorithm to construct the ***anti-Voronoi*** diagram of the vertices of P. An anti-Voronoi diagram (also known as a furthest-point Voronoi diagram) partitions the plane into regions, each containing the points *further* from one vertex of P than any other. The dual *anti-Delaunay triangulation* is the unique triangulation of the vertices of P where *none* of the edges are locally Delaunay. *[Hint: This should also be fairly easy.]*

(c) Describe and analyze an algorithm to construct the ***medial axis*** of P. The medial axis partitions the plane into regions, each containing the points closer to one *edge* of P than any other edge. Because P is convex, the medial axis is a planar straight-line tree; each edge of the medial axis lies on an *angle* bisector of two edges of P.



The Voronoi diagram, anti-Voronoi diagram, and medial axis of a convex polygon.

5. There is no problem 5. *(I will drop an extra homework score in my final grade computation.)*