Advanced Algorithms: Homework 7

Due on Mar. 27, 2024 at 11:59pm EST

Professor Dana Randall Spring 2024

As stated in the syllabus, unauthorized use of previous semester course materials is strictly prohibited in this course.

Sarthak Mohanty

Exercise 1

It is sometimes of interest to compute the Voronoi diagram of a set of sites, but we are only interested in a portion of the final diagram. In this problem, we'll consider how to compute the Voronoi diagram of a set of points in $\mathbb{R}^2$, but restricted to a given line ℓ. By rotating and translating space, we may assume that ℓ is aligned with the x-axis.

You are given a sequence of n sites in the plane $P = \{p_1, \ldots, p_n\}$ sorted in increasing order of their x-coordinates (see Fig. 1(a)). Present an algorithm that computes the Voronoi diagram of P, but restricted only to the x-axis. (We don't care about the portion of the diagram lying above or below the axis.)

Observe that the diagram is a sequence of intervals that subdivide the x-axis. The output consists of a sequence of (at most $n-1$) endpoints of the segments $\{x_1, \ldots, x_m\}$, and each edge is labeled with the index of the associated site corresponding to this interval (see Fig. 1(b)). Your algorithm should run in $O(n)$ time. (*Hint:* Start by proving that the left-to-right order of the labels along the x-axis is consistent with the left-to-right order of the sites.)

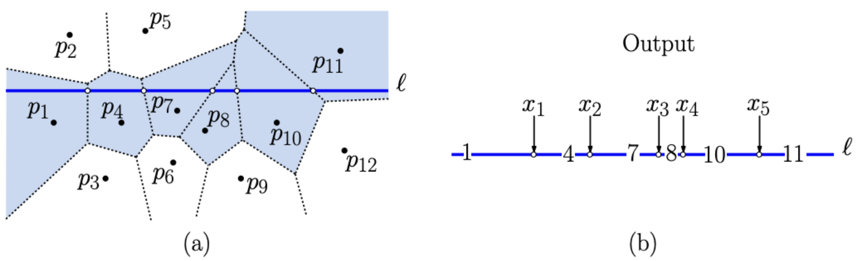

Figure 1: Restriction of a Voronoi diagram to a line.

Exercise 2

(Erickson) Let P be any set of points in the plane. A triangulation of P is a planar straight-line graph whose outer face is the complement of the convex hull of P, and whose bounded faces are all triangles (see Figure 2). There are several different ways to measure the *quality* of a triangulation. The goal of this question

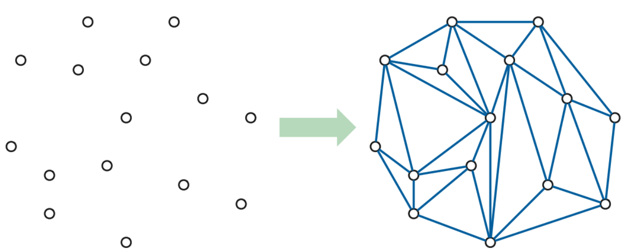

Figure 2: An example of a triangulation

is to prove that the Delaunay triangulation of P is the best possible triangulation of P, for a few different definitions of "best".

For example, Lecture 17 of David Mount's notes[1] contains a proof that among all triangulations of P, the Delaunay triangulation of P has the largest minimum angle.

[1] I gave this as part of your lecture notes.

(a) Let T be any triangulation of P. Prove that if every interior angle of T is acute, then T is the Delaunay triangulation of P.

(b) For any triangle Δ with vertices pqr, define

$$\text{Vol}(\Delta) = \text{area}(\Delta) \cdot (\|p\|^2 + \|q\|^2 + \|r\|^2)$$

where $\|(a,b)\|^2 = a^2 + b^2$ is the squared Euclidean norm. For any triangulation T, let $\text{Vol}(T) = \sum_{\Delta \in T} \text{Vol}(\Delta)$. Prove that among all triangulations of P, the Delaunay triangulation of P minimizes $\text{Vol}(T)$. (*Hint:* Why is the function called "Vol"?)

Exercise 3

Consider the following algorithm for VERTEXCOVER, run DFS, then output the nodes which are not leaves in the DFS tree. Show that the output is indeed a vertex cover, and that this algorithm gives yet another 2-approximation for the minimum vertex cover.

Exercise 4

We claimed in class that a bad approach to approximating the optimal vertex cover is to use a greedy method that does the following: pick the vertex that covers the most yet uncovered vertices, add it to the cover, and repeat. Give an infinite family of examples (i.e., one for each value of n) that shows that this method will not achieve an approximation ratio of 2.

Exercise 5

We saw a version of the load-balancing problem (on the exam) where we had 2 machines. Now we have m machines $M_1 \dots M_m$ and a set of n jobs. Each job j has a processing time t_j. We seek to assign each job to one of the machines so that the loads placed on all machines are as "balanced" as possible. That is, if $A(i)$ are the set of jobs assigned to machine M_i, then the load for machine M_i is $T_i = \sum_{j \in A(i)} t_j$. We want to minimize the *makespan* $T = \max_i T_i$. This problem is NP-complete, so instead you should design an approximation algorithm for the problem. What approximation ratio does the greedy algorithm (assign each job to the least loaded machine) yield? Prove your answer.