Eidgenössische Technische Hochschule Zürich
Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Swiss Federal Institute of Technology Zurich

Institute for Theoretical Computer Science
Dr. B. Gärtner, A. Gundert and Dr. M. Hoffmann

11.10.2012

Computational Geometry Homework 1 HS12

URL: http://www.ti.inf.ethz.ch/ew/courses/CG12/

Exercise 1 (20 points)

Suppose we are given a set P of n points in the plane $\mathbb{R}^2$, an additional point q of which we know that it is contained in $\text{conv}(P)$ and a ray $\mathcal{R}$ starting at q. The goal of this exercise is an algorithm that computes in $O(n)$ time the edge or the vertex of the boundary of $\text{conv}(P)$ at which $\mathcal{R}$ leaves $\text{conv}(P)$. Note that we are not given $\text{conv}(P)$ and also can't compute it in $O(n)$ time.

a) We first consider an auxiliary procedure which we will apply later. Let S be a set of $2n$ points such that none of these points lies on the x-axis. Suppose we are given a perfect matching $M = \{m_1, m_2, \ldots, m_n\}$ of S such that for each m_i one of its endpoints lies above and the other below the x-axis. Our procedure $\textsc{Bowtie}(M)$ then computes an edge $e = e_n$ as follows: $e_1 = m_1$, $e_{i+1} = \ltimes(e_i, m_{i+1})$. Here for two edges e, e' crossing the x-axis $\ltimes(e, e')$ denotes the edge of the boundary of $\text{conv}(e \cup e')$ that crosses the x-axis at the smallest x-coordinate.

Show that for any directed line ℓ that points upwards and crosses the x-axis left of the crossing with the edge e computed by $\textsc{Bowtie}(M)$ at least half of the points of S lie to the right of ℓ.

b) Back to our original problem. Assuming w.l.o.g. that $\mathcal{R}$ is horizontal and points to the left of q, we first describe the basic idea of our algorithm. Repeatedly do the following: Choose a point $p_0 \in P$ that lies above $\mathcal{R}$. Compute the right tangent t of p to $\text{conv}(\{q\} \cup \{p \in P : p \text{ is below } \mathcal{R}\})$ ("right" as seen from p_0) and discard all points that lie to the right of t and above $\mathcal{R}$.

What is the running time of a single such step? Why and how does this help to compute the intersection with the boundary of $\text{conv}(P)$? Sketch a correct algorithm based on this idea that can still have a worse than linear runtime. What is the runtime of your algorithm?

c) Simply choosing the points p_0 above $\mathcal{R}$ in an arbitrary order might not be the best idea. Here is an idea for a smarter way to pick our point:

Let A be the set of points above $\mathcal{R}$ and let p_m be the median of points in A w.r.t. the order in which the points can be seen from q. Choose a perfect matching M of the points to the left and to the right of the ray from q through p_m and apply $\textsc{Bowtie}$ to get an edge e. We claim that one of the endpoints of e is a good choice for p_0, i.e., we can discard sufficiently many points.

Describe the algorithm in detail and show that we can achieve an overall runtime of $O(n)$.

Exercise 2 (10 points)

A polygon P is called *vertically connected* if the intersection $\ell \cap P$ with any vertical line ℓ has at most one component. Show that an n-vertex vertically connected polygon, given as a list of its vertices in counterclockwise order, can be triangulated in time $O(n)$.

Exercise 3 (30 points)

Choose one of the topics below to investigate. Find the relevant research papers, surveys or textbooks that deal with this problem and find out what is known about it. What are the main results? What are the open questions related to this problem? You are not supposed to read papers in detail, but rather to try to gain an overview. Hand in a short report (of between 1 and 2 pages) about the problem and what you have found out. Your report should contain:

- an informal description of the problem using your own words (as you would explain it to a friend),
- a precise definition of the problem (in which even the most nitpicking reader is not able to find anything unclear),
- the important results regarding this problem (providing enough explanations to make the difference between the results apparent, but without going into unnecessary details),
- the current state of the problem (how much has already been solved, what remains open),
- a complete list of references (every theorem or result you state should have an appropriate citation so that it is easy to find [and check]; if you have fewer than 3 or 4 references, you might have searched not thoroughly enough).

Note that you can access many journals from the ETH network only. If you want to search at home, you might need to connect to the ETH intranet using VPN.

a) Minimum weight triangulation.

b) Counting planar triangulations.

c) Additively weighted Voronoi diagrams.

d) Higher order Delaunay triangulations.

e) Halfplane range searching.

f) Order types of point sets.

g) Pseudo-triangulations.

h) Delaunay refinement meshing.

i) Kinetic data structures in geometry.

j) ILP (integer linear programming) and relaxation.

k) Linear programming with few violated constraints.

l) Support vector machines.

m) Geometric in-place-algorithms.

n) Well-separated pair decomposition.

o) Core sets.

The choice of your topic should be briefly discussed with the assistant, so that no topic is assigned twice.

Due date: 25.10.2012, 13h15