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, Dr. M. Hoffmann and Tobias Christ

1.10.2009

Computational Geometry — Homework 1 — HS09

URL: http://www.ti.inf.ethz.ch/ew/courses/CG09/

Exercise 1 (10 points)

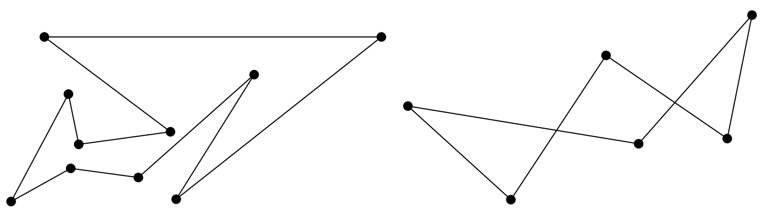
In this exercise, you are asked to think about some issues related to local vs. global convexity. These issues are related to the correctness of the Graham Scan algorithm, an aspect that was deliberately swept under the rug during the lecture.

a) Let $P = (p_0, \ldots, p_{n-1})$ be a sequence of n pairwise distinct points in $\mathbb{R}^2$. Prof. Mac Easy claims that you can check by means of the following algorithm whether or not P describes the boundary of a convex polygon in counter clockwise order:

```
bool is_convex(p_0, ..., p_{n-1}) {
    for (int i = 0; i <= n - 1; i = i + 1)
        if (rightturn(p_i, p_{(i+1) mod n}, p_{(i+2) mod n}))
            return false;
    return true;
}
```

Disprove this claim and correct the algorithm.

b) We have seen in the lecture that computing the convex hull of a set of n points in $\mathbb{R}^2$ requires $\Omega(n \log n)$ geometric operations. The situation changes, however, if the input is not an unstructured set, but, for instance, a simple polygon (a region bounded by a closed polygonal chain that does not intersect itself; see the figure below for an example). Prof. Mac Easy claims that in this case the convex hull can be computed in linear time as follows.



(a) A simple polygon. (b) A non-simple polygon.

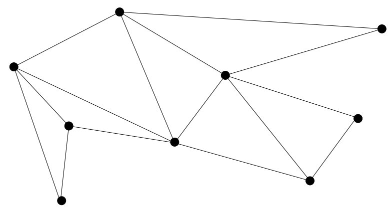
Input: $n \in \mathbb{N}$, and a sequence $(p_1, \ldots, p_n)$, such that $p_i \in \mathbb{R}^2$, for $1 \le i \le n$, and $(p_1, \ldots, p_n, p_1)$ forms the boundary of a simple polygon.

Algorithm: Run the second phase of Graham Scan (Successive Local Repair) on $(p_1, \ldots, p_n)$. As seen in the lecture, this computes the convex hull and requires $O(n)$ geometric operations only.

Show that the claim is correct whereas the proof is wrong. More precisely, describe a polygon—preferably with as few vertices as possible—for that the above algorithm fails to compute the convex hull. Then sketch how to fix the algorithm such that it indeed provides a linear time algorithm to compute the convex hull for a simple polygon. Finally, argue why the Graham Scan algorithm as presented in the lecture is correct.

Exercise 2 (10 points)

Let P be a simple polygon. A triangulation of P is a triangulation whose vertices are the vertices of P, and whose outer face is formed by the edges of P. Here is an example of a simple polygon triangulation:



a) Prove that every simple polygon P has a triangulation.

b) Find for every $n \geq 3$ a simple polygon P with n vertices that has exactly one triangulation. P should be in general position, meaning that no three of its vertices are collinear.

Exercise 3 (30 points)

Choose one of the problems below to investigate. Find the relevant research papers that deal with this problem and find out what is known so far about it. What are the main results? What are the open questions related to this problem? You are not supposed to read the papers in detail, but try to gain an overview. (If you work from somewhere outside ETH, connect to the ETH intranet using VPN to gain access to the different online journals, where you can find the relevant material.) You are supposed to hand in a small report (between 1 and 2 pages) about the problem and what you found out. You 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 that's not clearly stated/explained)
- a chronological list of the important results regarding this problem (providing enough explanations to make the difference between the different results apparent, but without going into details),
- the current state of the problem (in how far the problem has been solved or what questions still are open),
- a complete list of references (if there are fewer than 3 or 4, you probably did not research enough).

a) Minimum weight triangulation of a point set in the plane.

b) Output-sensitive convex hull algorithms in the plane ($O(n \log h)$).

c) (Semi-)Dynamic convex hull in the plane (maintenance of the convex hull under point insertion and/or deletion).

Due date: 15.10.2009, 13h15