2. Assignment on

Computational Geometry

Panos Giannopoulos, Wolfgang Mulzer, Lena Schlipf SoSe 2013

Due on 22.April 2013 in the tutorial session

Problem 1 Convexity *10 points*

(a) Let $\{C_i\}_{i\in I}$ be a collection of convex sets in the plane. Show that the intersection $\bigcap_{i\in I} C_i$ of these sets is convex. Is there a similar property for the union of the sets?

(b) Let P be a finite point set in the plane. Show that the boundary of the convex hull $\mathrm{CH}(P)$ of P is a convex polygon whose vertices are points of P.

(c) Show that the segment between two points $p, q \in P$ is an edge of $\mathrm{CH}(P)$ if and only if all points of P lie on the same side of the line through p and q.

Problem 2 Computing the tangents to a polygon *10 points*

In the lecture, we discussed Chan's algorithm which computes the convex hull, $\mathrm{CH}(P)$, of an n-point set P. In the main procedure $\textsc{hull}(P, h^*)$ which computes the first h^* points of $\mathrm{CH}(P)$, in particular in Line 5(i), we need to compute the point tangent $q_i \in \mathrm{CH}(P_i)$ that maximizes the angle $\angle p_{k-1}p_kq_i$. Show that this can be computed in $O(\log h^*)$ time. (Recall that the size of each subset P_i is at most h^*.) In other words, show that given a convex polygon (as an array of its vertices in ccw order) with h^* vertices and a point outside the polygon, one can compute the two tangents (also called supporting lines) from the point to the polygon in $\log h^*$ time.

Problem 3 Chan's algorithm and superexponential search *10 points*

Consider again Chan's algorithm.

(a) Adapt the part of $\textsc{hull}(P, h^*)$ that computes the first h^* points of $\mathrm{CH}(P)$ so that it takes $O(n)$ time, instead of $n \log h^*$ time like we discussed in the lecture. Note that the total running time will not be affected by this change. (*Hint:* the points of tangency q_i on the small hulls $\mathrm{CH}(P_i)$ advance in ccw direction.)

(b) Show that by running $\textsc{hull}(P, h^*)$ for $h^* = 2^{2^t}$ where $t = 0, 1, 2, \ldots$, the running time of the final algorithm is $\mathcal{O}(n \log h)$.