Computational Geometry, SS 2020
Assignment 1
Due: Wednesday 13.5, 8:00 (morning!) CET

Send your solutions in pdf format to:

`andre.nusser@mpi-inf.mpg.de`

You can discuss about the exercises in groups (respecting social distancing, obviously), but you have to hand in a solution *in your own words*. Note that, while there are a lot of exercises on this sheet, you just have to obtain "20 · #exercise sheets" points during the semester to be admitted to the exam. You can choose which exercises you want to try yourself on.

Problem 1. (6 points)
Consider $\mathbb{R}^3$. For line ℓ through the origin let $\mathcal{N}(\ell)$ be the unique plane through the origin that is normal to ℓ. Analogously, for a plane h through the origin we let $\mathcal{N}(h)$ be the unique line through the origin that is normal to h. We will refer to a pair $\{\ell, h\}$ where ℓ is a line passing through the origin and h is a plane passing through the origin with $h = \mathcal{N}(\ell)$ (and therefore also $\ell = \mathcal{N}(h)$) as a *"thumbtack"*.

In class we discussed how thumbtacks can be used to realize point–line duality transforms in the plane by considering the intersection of parts of a thumbtack with two planes H, I that do not contain the origin.

1. Verify the claim made in class that the first duality discussed in class ($ax+by = 1 \longleftrightarrow (a, b)$) is described by using the planes $H = \{ (x, y, 1) \mid x, y \in \mathbb{R} \}$ and $I = \{ (\xi, \eta, -1) \mid \xi, \eta \in \mathbb{R} \}$.
2. Now consider the second duality that we discussed in class ($y + d = kx \longleftrightarrow (k, d)$). If you use H as before, what would be the Plane I that realized this duality?
3. We briefly discussed in class that point–line duality transforms in the plane have "exceptions," i.e. points and lines that are not mapped to anything. How does this effect manifest itself in the thumbtack view?

Problem 2. (6 points)
Let L be a set of n lines in the plane and let $\mathcal{A} = \mathcal{A}(L)$ be the arrangement of L. For a cell c of $\mathcal{A}$ let $\deg(c)$ be the number of edges bounding c.

1. Prove the following:
$$\sum_{c \text{ cell of } \mathcal{A}} \deg^2(c) = O(n^2)$$
2. You can view $\mathcal{A}$ as a planar graph with $O(N)$ vertices, edges, and faces, where $N = n^2$. (Topologically you could remove the unboundedness by bending all unbounded edges to end in a common vertex.) The previous question shows

that for the planar graph G formed by an arrangement of combinatorial size N you have

$$\sum_{f \text{ face of } G} \deg^2(f) = O(N)\,.$$

Show that such a statement does not hold for planar graphs in general. How large can you make

$$\sum_{f \text{ face of } G} \deg^2(f)$$

when G has $O(N)$ vertices, faces, and edges?

Problem 3. (8 points)

Let S be a set of n points in the plane, and let k be such that $0 \le k \le n$, and let ℓ be a non-vertical line. We call ℓ *a strict k-separator for S* iff the open halfplane below ℓ contains exactly k points of S and the halfplane above ℓ contains exactly $n-k$ points of S. We call ℓ *an anchored k-separator for S* iff the open halfplane below ℓ contains exactly k points of S and the halfplane above ℓ contains fewer than $n-k$ points of S.

1. Describe the structure of all strict and of all anchored k-separators for S in the dual.
2. A *halving line* for S is a line that contains the same number of points of S in each of the two open halfplanes bounded by the line.

   Give a proof of the ham-sandwich theorem for points sets, which says that for any set R of r red points and any set B of b blue points there is a line that is simultaneously a halving line for R and for B.
3. Show that such a simultaneously halving line can be constructed in $O(r^2 + b^2)$ time.

Problem 4. (10 points, extra credit)

Let P be a convex 3-dimensional polytope with n facets. For each direction a in $\mathbb{R}^3$ let $S(a, P)$ be the area of the projection P onto a plane normal to a (the area of the "shadow" of P).

Give an algorithm that finds the directions a that minimize/maximize $S(a, P)$. Your algorithm should run in $O(n^2)$ time.

Hint: For a facet F of P let n_F be the outward normal vector of F with length equal to the area of F. Then the area of the projection of F in unit direction a is essentially given by the inner product $\langle n_F, a\rangle$.

Problem 5. (5 points)

Let p and $(q_1, \ldots, q_n)$ be points in $\mathbb{R}^2$, where $q_1, \ldots, q_n$ are the vertices of a convex polygon C in clockwise order, and p is outside C. Compute the tangent lines from p to C in $O(\log n)$ time, assuming that retrieving the coordinates of any q_i can be done in constant time.

Problem 6. (5 points)
Show that an $O(n + h)$ Real-RAM algorithm for convex hull in $\mathbb{R}^2$ implies that number sorting can be done in $O(n)$ time. Your Real-RAM machine only has basic arithmetic functions (addition, subtraction, multiplication, division), i.e., no square root or trigonometric functions.

Problem 7. (8 points)
Let P_1 and P_2 be two convex polygons with n vertices in total. Give an $O(n)$ time algorithm that computes the convex hull of $P_1 \cup P_2$. Use this algorithm as a subroutine to create an $O(n \log n)$ divide-and-conquer algorithm for convex hull in $\mathbb{R}^2$.

Problem 8. (10 points)
Let C be a convex polytope in $\mathbb{R}^3$ with vertices $p_1, \ldots, p_n$ and edge graph G, and let T be a triangulation of G. Given the coordinates of $p_1, \ldots, p_n$ and an unsorted list of the $3n - 6$ faces of T (where face $p_i p_j p_k$ of T is described with the unsorted triplet (i, j, k)), compute a doubly-connected edge list[1] representation of G in $O(n)$ time, using
(a) $O(n^2)$ addressable memory slots
(b) $O(n)$ addressable memory slots.

(Try not to get bogged down in detailed manipulation of pointers.)

[1] In a doubly connected edge list, each edge of G is represented by two opposing arcs (sometimes called half-edges) that are associated with the two incident faces: each face is described by a cycle of half-edges that have a counter-clockwise orientation when looked at from outside C. Each half-edge has a pointer to the opposing half edge, and to the previous and next half-edge of its face.