Assignment no. 3

due: May 1st, 2019

Exercise 3.1 The *stabbing number* of a triangulation of a simple polygon P is the maximum number of diagonals intersected by any line segment interior to P. Give an algorithm that computes a triangulation of a convex polygon that has stabbing number $O(\log n)$.

Exercise 3.2 In each of the following settings, give an example where the specified **vertex guards** do not fully cover the polygon in the art gallery sense:
(a) A simple polygon with $2k$ vertices, for every $k > 2$, and a specific assignment of guards placed at **every other vertex** along the boundary of the polygon. Namely, guards placed at the vertices $v_i, v_{i+2}, v_{i+4}, \ldots,$ do not fully cover the polygon.
(b) Similarly, a simple polygon with $3k$ vertices, for every $k > 2$, and a specific assignment of guards placed at **every third vertex** along the boundary of the polygon.
(c) A simple polygon with n vertices, for every $n > 5$, and guards placed only at **convex vertices**. A vertex is convex if its interior angle is less than π.

Exercise 3.3 On n parallel railway tracks n trains are going with constant speeds $v_1, v_2, \ldots, v_n$. At time $t = 0$ the trains are at positions $k_1, k_2, \ldots, k_n$. Give an $O(n \log n)$ time algorithm that detects all trains that at some moment in time are leading.

Exercise 3.4 Instead of removing the object from the mold by a single translation (as we saw in class), we can also try to remove it by a single rotation. For simplicity let's consider the planar variant of this casting problem, and let's only look at clockwise rotations.
(a) Give an example of a simple polygon P with top facet f that is not castable when we require that P should be removed from the mold by a single translation, but that is castable using rotation around a point.
(b) Show that the problem of finding a center of rotation that allows us to remove P with a single rotation from its mold can be reduced to the problem of finding a point in the common intersection of a set of half-planes.
(CGAA Ex. 4.7)

Exercise 3.5 Describe in detail the procedure **UnboundedLP3**, whose input is an objective function and a set of half-spaces in $\mathbb{R}^3$. The procedure decides whether the underlying linear program is unbounded. If it is, the procedure computes a ray in the feasible region such that the objective improves as we proceed along the ray away from its terminus. If the program is bounded, the procedure returns three of the input half-spaces as witnesses to this effect. Analyze the running time of the procedure.
No need to prove the correctness of the procedure.
No need to provide the procedure for the bounded case!

Notice that there are additional exercises on the other side of the page.

Exercise 3.6 (self-study, do not submit) Acquaint yourself with the deterministic linear-time algorithm for solving two-variable linear programs by Meggido. It is clearly described in Section 7.2.5, Two-variable linear programming, of the Computational Geometry book by Preparata and Shamos, the 1985 Edition.

Exercise 3.7 (optional, bonus) The *pockets* of a simple polygon are the areas outside the polygon, but inside its convex hull. Let P_1 be a simple polygon with n_1 vertices, and assume that a triangulation of P_1 as well as of its pockets is given. Let P_2 be a convex polygon with n_2 vertices. Show that the intersection $P_1 \cap P_2$ can be computed in $O(n_1 + n_2)$ time. (CGAA Ex. 3.12)
Remarks. (1) The fine details matter—provide a precise description of your solution. (2) You may assume that the triangulation of P_1 together with the triangulation of its pockets, is given as a DCEL.