Assignment no. 3

due: Monday, April 25th, 2022

Exercise 3.1 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 3.2 Let H be a set of at least three half-planes in the plane with a non-empty intersection such that not all bounding lines are parallel. We call a half-plane $h \in H$ redundant if it does not contribute an edge to $\bigcap H$. Prove that for any redundant half-plane $h \in H$ there are two half-planes $h', h'' \in H$ such that $h' \cap h'' \subset h$. Give an $O(n \log n)$ time algorithm to compute all redundant half-planes in H.

Exercise 3.3 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 3.4 Give an example of a set of n points in the plane, and a query rectangle for which the number of "grey" nodes of the kd-tree visited is $\Omega(\sqrt{n})$, namely the overhead term in the query time is $\Omega(\sqrt{n})$.

Exercise 3.5 The algorithm we saw in class for searching in a kd-tree (where the search is guided by comparing the *region* of a node with the query region) can also be used when querying with ranges other than rectangles. For example, a query is answered correctly if the range is a triangle.
(a) Show that the query time for range queries with triangles is linear in the worst case, even if no points are reported at all. Hint: Choose all the input points to lie on the line $y = x$.
(b) Suppose that a data structure is needed that can answer triangular range queries but only for triangles whose edges are horizontal, vertical or have slope $+1$ or -1. Devise a linear-size data structure that answers such queries in $O(n^{3/4} + k)$ time, where k is the number of points to be reported. Hint: Choose 4 coordinate axes in the plane and use a "4-dimensional" kd-tree.
(c) Improve the query time to $O(n^{2/3} + k)$.

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

Exercise 3.6 (optional) Given a three-dimensional linear program, describe a procedure to find three witness half-spaces to the program's boundedness, if indeed it is bounded.

Exercise 3.7 (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.