Assignment no. 4

due: June 1st, 2020

Exercise 4.1 Let L be a set of n lines in the plane. Give an $O(n \log n)$ time algorithm to compute an axis-parallel rectangle that contains all the vertices of the arrangement $\mathcal{A}(L)$ in its interior.

Exercise 4.2 Hopcroft's problem is to decide, given n lines and n points in the plane, whether any point is contained in any line. Give an $O(n^{3/2} \log n)$ time algorithm to solve Hopcroft's problem. Hint: Give an $O(n \log n)$ time algorithm to decide, given n lines and $\sqrt{n}$ points in the plane, whether any point is contained in any line.

Exercise 4.3 Let S be a set of n segments in the plane. A line ℓ that intersects all the segments of S is called a transversal or stabber for S.
(a) Give an $O(n^2)$ algorithm to decide if a stabber exists for S.
(b) Now assume that all segments are vertical. Give a randomized algorithm with $O(n)$ expected running time that decides if a stabber exists for S. (CGAA Ex. 8.16)

Exercise 4.4 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 4.5 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 $O(n^{2/3} + k)$.