Computational Geometry, SS 2020
Assignment 3
Due: Wednesday 10.6, 8:00 (morning!) CET

Send your solutions in pdf format to:

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

Problem 1. (6 points)
A point $(x, y) \in \mathbb{R}^2$ *dominates* (x', y') if $x > x'$ and $y > y'$. Two points are *incomparable* if neither point dominates the other. Preprocess a set P of n points into a data structure which can report in $O(\log n + k)$ time the points in P that are incomparable to a query point (where k is the number of reported points). Your data structure should fit in $O(n)$ space.

Problem 2. (10 points)
Given a set S of disjoint spheres in $\mathbb{R}^3$ with their center and radius, describe a data structure that can determine the first sphere hit by a vertical ray (i.e., parallel to the z-axis) shot up from a query point in $O(\log n)$ time. The data structure should fit in $O(n^3)$ space.

Problem 3. (8 points)
Develop a sweep algorithm that given a set S of axes-aligned rectangles in the plane (i.e. Cartesian products of intervals) determines the number of pairs of rectangles in S that intersect. Be sure to state the invariants of your sweep and the nature of your sweepline structure and event queue. What running time and space usage can you achieve?

Problem 4. (8 points)
Develop a data structure that processes a set S of axes-aligned rectangles so that queries of the following form can be answered quickly: Given an axes-aligned query rectangle Q what is the area of $Q \cap \bigcup S$?

1. Can you find a structure that achieves logarithmic query time with, say, quadratic space?
2. What kind of query time can you achieve with $O(n \cdot \text{polylog}(n))$ space?

Hint: Things may be simpler if you "decompose" each query into simpler queries, all of the same form.

Problem 5. (8 points)
In class we discussed a method for planar point location (or vertical ray shooting among non-crossing x-monotone segments) that was based on segment trees. It processed a query point q by first determining $\text{path}(q.x)$ in the segment tree (using x-comparisons) and then searching for q in each S_v for vertex v on $\text{path}(q.x)$. It achieves a query time $Q(n) = O(\log^2 n)$ with space $S(n) = O(n \log n)$ and preprocessing $P(n) = O(n \log n)$.

Earlier in the course we discussed fractional cascading. It is a method that speeds up searches in sequences of ordered lists by copying (or moving?) every c^{th} element from on list to the next, where c is some constant.

Try to apply fractional cascading in this context.

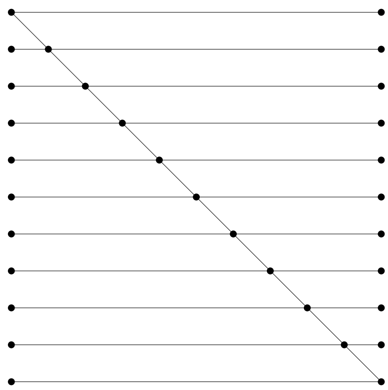
1. Can you achieve $Q(n) = 2 \cdot \log_2 n + O(1)$?
2. Can you achieve $Q(n) = 3 \cdot \log_2 n + O(1)$ while keeping $S(n) = O(n \log n)$?
3. Can you achieve $Q(n) = a \cdot \log_2 n + O(1)$ for some $a < 3$ while keeping $S(n) = O(n \log n)$?

For the purposes of this question $Q(n)$ counts the number of x-comparisons and y-comparisons (i.e. above/below segment comparisons) in a worst case query.

Problem 6. (6 points)

For the purposes of this exercise a *standard model query structure* for vertical ray shooting in a set S of non-crossing x-monotone segments is a directed acyclic graph G with one source. Each sink in G is labelled with a segment in s (the answer when arriving at that sink); each non-sink node has outdegree 2 with one outgoing edge labelled with "yes" and the other with "no;" each non-sink node is either labelled as an x-node, in which case it has an x-comparison key that is the x-coordinate of some endpoint of a segment in S, or it is labelled as a y-node, in which case it has y-comparison key that is a segment $s \in S$. Queries are executed in the obvious way.

Prove that any standard model query structure G for the set S of $3n$ segments sketched below so that G is a tree must have at least $\Omega(n \log n)$ nodes.



Problem 7. (no points – just something to think about)

Let $G = (V, E)$ be a graph with n vertices and m edges. Prove that G must contain an independent set of vertices of size at least $\sum_{v \in V} \frac{1}{1+\deg(v)}$, and show that such a set can be found in linear time.

Prove that the stated bound is never smaller than $\frac{n}{1+\overline{d}}$, where $\overline{d}$ is the average degree in G.