Applied Computational Geometry — Spring 2009 — Dan Halperin

Assignment no. 3

due: May 11th, 2009

Notice that all the exercises ask for *exact* output; all non-integer input is assumed to be given as rational numbers.
As before, you'll find additional useful information concerning input/output files, the way to operate the program, the submission of exercises and more in the course's website. Notice that it is important to comply with the rules of submission, to enable uniform testing of your programs.

Exercise 3.1: Nearest Jeep over time (30 points)
A group of Jeeps are driving in a flat desert, each in a straight lane and fixed velocity (the lanes or velocities are not necessarily the same, and the lanes may cross). They all start and stop simultaneously. We are given the start and stop coordinates for each Jeep. The base station is located at the origin. Write a program that finds the nearest Jeep to the base station over time. That is, the output should be a 1-dimensional arrangement that represents time, where each edge (resp. vertex) is associated with the nearest Jeep (or Jeeps) at that time segment (resp. point).
(optional) Give visual evidence (through simulation) to the correctness of your solution.

Exercise 3.2: Intersection of circles (20 points)
In this exercise, the input for each circle is given as three rational numbers: the coordinates of the center, and the radius. Notice that the decision procedures need to be exact, and only the reported coordinates may be approximate.
(a) Read the data of two distinct circles in the plane. Decide whether the circles are tangent, intersect in two points, or are disjoint. If they intersect, print out an approximation of the intersection point(s).
(b) Read the data of three distinct circles in the plane. Decide whether the three circles meet in a single point (YES or NO). If they do, print out an approximation of their common intersection point.
Use algebraic number types (possibly CORE's), but otherwise devise and implement your own solution.

Exercise 3.3: Largest common point sets under ε-congruence (50 points)
This is a fairly advanced exercise. Feel free to use as much ready-made software as possible (with the exception of a full-fledged solution to the entire problem). For example, you can use the Boost Graph Library to find the maximal cardinality matching in bipartite graphs.
Given two finite sets of points A and B in the plane, write a program that finds equally sized subsets $A' \subseteq A$ and $B' \subseteq B$ of maximal cardinality such that points in A' match points in B' under translation, up to distance at most some given ε. Namely, each point a in A' has a unique point b in the translated B' (and vice versa) such that the Euclidean distance between a and b is at most ε and A' is the largest cardinality such subset.
In the course's website you will find solution hints. We encourage you to try and come up with your own solution before looking at this extra information.