Algorithms, Probability, and Computing Fall 2011
Exercise Sheet 4

General rules for solving exercises

- This is a theory course, which means: if an exercise does not explicitly say "you do not need to prove your answer" or "justify intuitively", then a formal proof is always required.
- All exercises and their solutions, no matter whether they are graded or regular/optional ones, are part of the material relevant for the two exams.
- The tasks marked with the symbol * are more difficult than usual exercises, either because they require a non-standard idea or involve some non-trivial calculation. This is merely a help for you to decide which exercises you would like to spend how much of your time on. Note that all exercises and their solutions count as part of the material relevant for the two exams.

The following exercises will be solved in the exercise classes of October 19/21.

Exercise 1 - Lines Intersecting Segment
Solve Exercise 2.5 from the lecture notes.

Exercise 2 - Below a Line, in Convex Position
Solve Exercise 2.9 from the lecture notes.

Exercise 3 - Number of Cells in Arrangements
Solve Exercise 2.11 from the lecture notes.

Exercise 4 - Not too Many, not too Few
Solve Exercise 2.15 from the lecture notes.

Exercise 5 - Nearest Neighbor Changes
Solve Exercise 2.17 from the lecture notes.

Exercise 6 - Farthest Point Voronoi Diagrams
You want to open a restaurant – let us call it “Rose’s” – that will have its food delivered to customers by bike messengers. This is a somewhat demanding endeavor, as you need to avoid the worst case that the food gets cold before it reaches the customer; i.e., you want to keep an eye on the largest distance to a customer. You already have the coordinates of your most important future customers. In order to make an informed decision on where to rent a place for your business, you want to draw a map that lists for all places in the city the farthest future customer. Assume you do not live in Zurich but in a city where Euclidean distances are a valid approximation of biking times.

As you certainly already have guessed, a locus approach is advisable here. We abstract the customers to a set $S$ of $n$ points in the plane, out of which no three lie on a common line and no four on the same circle. The structure we are looking for goes by the name of farthest point Voronoi diagram. It divides the plane into regions that have the same farthest point in $S$.

1. Express $V^F_p$, the farthest point Voronoi cell of $p$, i.e. the set of points in the plane for which $p$ is the farthest point in $S$, in terms of $h(p,p')$ as defined in the lecture notes on Page 18 of Chapter 2.
2. Unlike $V_S(p)$, $V_{S_b}^F(p)$ does not necessarily contain a point. Give a small example $S_b$ of a point set and a point $p \in S_b$ for which $V_{S_b}^F(p)$ is empty.

3. Although you do not live in Zurich, the search for an affordable place for your restaurant drags on and you need to evaluate a lot of places. To make your life easier, you want to implement the look-up of farthest customers (instead of consulting the diagram yourself). Employing the technique encountered in Section 2.4 of the lecture notes, what asymptotic query time can you achieve? In order to support your answer, describe in what points you have to adapt the algorithm or its analysis found in the lecture notes. You do not need to give a lower bound to prove the optimality of your answer.

**Hint:** Think about query time in function of the complexity of the farthest point Voronoi diagram first in order to find out how sharp your analysis of the complexity of the diagram needs to be.