Computational Geometry—236719

(Spring 2019—Gill Barequet and Gil Ben-Shachar)

Assignment no. **2**

Given: May 13, 2019
Due: **May 27, 2019**

Submission in **singletons**

Question 1.
Let H be a *finite* set of at least three half-planes. 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 $h' \cap h'' \subset h$.

Question 2.
In some applications one is interested only in the number of points that lie in a range, rather than in reporting all of them. Such queries are often referred to as *range-counting* queries. In this case, one would like to avoid paying the $O(k)$ term in the query time.

1. Describe how a 1-dimensional range tree can be adapted such that a range-counting query can be performed in $O(\log n)$ time. Prove the query time bound.
2. Describe how d-dimensional range-counting queries can be answered in $O(\log^d n)$ time, using the solution to the 1-dimensional problem. Prove the query time.

Question 3.

1. Show that, given an unprocessed planar subdivision S (in a DCEL) with n vertices and edges, and a query point q, the face of S containing q can be computed in $O(n)$ time.
2. A convex polygon P is specified as an array of n vertices sorted along the boundary. Show that, given a query point q, it can be tested in $O(\log n)$ time whether or not q lies inside P.
3. A polygon P is called *star-shaped* if there exists a point p in the interior of P such that, for any other point q in P, the segment $\overline{pq}$ lies entirely in P. Generalize the solution of the convex case for star-shaped polygons. (You may assume that p is given.)

Question 4.
Assume that there are r points moving in the plane, each with a given constant speed and direction, denoted as *motorcycles*. Every motorcycle leaves a trail along the path it rides through. Motorcycles begin their ride simultaneously, and they ride until they either crash on the trail left

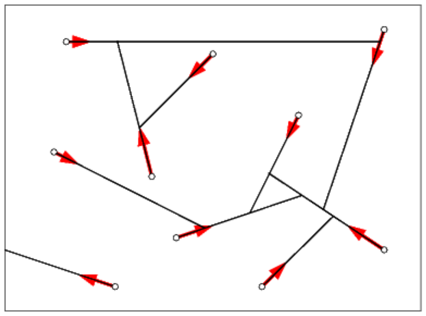

Figure 1: A motorcycle graph

by another motorcycle, or eventually run out of gas. (Assume that this happens only after all possible crashes occur.) Assume that no two motorcycles crash exactly unto each other. We define the *motorcycle graph* accordingly: The vertices are the initial positions of the motorcycles, crash points, and running-out-of-gas points, and the edges are the motorcycle traces (see Figure 1).

1. What is the complexity (number of vertices and edges) of the motorcycle graph? Prove your answer.
2. Describe a simple event-based algorithm for computing the motorcycle graph. Analyze the complexity of the algorithm.
3. Assume that the directions of all motorcycles have positive x components (i.e., all motorcycles "go east"). Can you describe A more efficient algorithm for computing the motorcycle graph?

Question 5.
Consider the Voronoi diagram of points lying on the surface of a sphere (where distances are measured on the sphere itself).

1. What is the bisector of two points? (Hint: consider symmetry. No need to prove.)
2. What is the curve that is equidistant from a point and a circle surrounding the point? (No need to prove.)
3. What is the degree of a Voronoi vertex, assuming that the point sites are in general position?
4. What is the combinatorial complexity of the diagram? (Provide exact bounds.)
5. Suggest an adaptation of the beach-line algorithm for computing the spherical diagram. Explain in detail.