Introduction to Computational Geometry, Fall 2013

Please type your homework and submit it before the start of class (3:30pm) Thursday, September 19, 2013. It can also be submitted by email, but must be sent at least one hour before the start of class (2:30pm) September 19, 2013.

Homework 1

1. The stereographic map $\Pi : \mathbb{R}^2 \to \mathbb{R}^3$ takes points in the plane and maps them onto a sphere. If we view the points in the plane as living on the plane $z = 1$ in $\mathbb{R}^3$, then the image of the a point $p = [x\ y\ 1]^\top$ is the point $\Pi(p)$ which is the intersection of line through p and the origin with the sphere centered at $[0\ 0\ \frac{1}{2}]^\top$ with radius $\frac{1}{2}$.

   One way to define Π is to observe that it is the restriction of the map

   $$x \in \mathbb{R}^3 \mapsto \frac{x}{\|x\|^2}$$

   to the plane $z = 1$. That is, for $p \in \mathbb{R}^2$,

   $$\Pi(p) = \frac{[p_x\ p_y\ 1]^\top}{p_x^2 + p_y^2 + 1}.$$

   A. Prove that this indeed gives the stereographic map.

   B. Show how the stereographic map can be used to compute inCircle predicates instead of the parabolic lifting used in class.

   C. Prove that your algorithm is correct.

2. Prove that the same parabolic lift trick shown in class works for inSphere tests in $\mathbb{R}^d$. The input is a set P of $d + 1$ points in $\mathbb{R}^d$ and a query point q in $\mathbb{R}^d$. The output is **YES** if q is inside the unique sphere circumscribing P and **NO** otherwise. What general position assumptions do you need to make this problem well-defined?

3. In class we gave an algorithm for testing intersections. It checked the orientations of 4 triangles. Prove that it works. Give an algorithm to test if a point is inside a closed triangle. Prove it is correct. Give an algorithm for testing if two triangles intersect.

4. Give an algorithm for testing if a given point is inside, outside, or on a convex polygon, given as a list of vertices in ccw order. Analyze the running time. You should be able to handle queries in $O(\log n)$ time. Does it work for non convex polygons? Sketch how it might be fixed for nonconvex polygons.

5. The area of a parallelogram is given by the determinant of two independent vectors (b-a) and (c-a) defining the sides. This also gives twice the area of the triangle (a,b,c). Use this to give an algorithm for the area of a convex polygon. Does this algorithm work for non convex polygons?

6. For the Graham Scan algorithm, the step of sorting the points by slope (or angle) gives an ordering that is a simple (i.e., non-self-intersecting) polygon. Show that this is not a sufficient condition for Graham scan to work by finding an example where Graham Scan fails if the points are ordered as they appear in a simple polygon rather than being sorted by angle. That is, you want to run the inner loop of the Graham Scan (the linear time part), but you are using th eorder given to you, not the sorted-by-angle ordering.

   Update: You should also assume that the first point is still the leftmost point. This avoids a trivial counterexample.

   Can you patch the algorithm? Just give a high level description. What geometric facts might you use? (Don't write more than four sentences for this last part)