

## COMPUTATIONAL GEOMETRY

### Homework Set # 2

Due at the beginning of class on Wednesday, March 2, 2011. *Reminder: Show your reasoning!*

---

Recommended Reading: BKOS: Chapter 2; O'Rourke sections 7.7, 7.8.

#### DO ANY 3 OF THE FOLLOWING 5 PROBLEMS.

(1). Problem 2.11, BCKO/BKOS. Let  $S$  be a set of  $n$  circles in the plane. Describe a plane sweep algorithm to compute all intersection points between the circles. (Because we deal with circles, not disks, two circles do not intersect if one lies entirely inside the other.) Your algorithm should run in  $O((n+k)\log n)$  time, where  $k$  is the number of intersection points.

(2). Problem 2.14, BCKO/BKOS. Let  $S$  be a set of  $n$  disjoint line segments in a plane, and let  $p$  be a point not on any of the line segments of  $S$ . We wish to determine all line segments of  $S$  that  $p$  can see, that is, all line segments of  $S$  that contain some point  $q$  so that the open segment  $\overline{pq}$  doesn't intersect any line segment of  $S$ . Give an  $O(n\log n)$  time algorithm for this problem that uses a rotating half-line with its endpoint at  $p$ .

(3). Problem 2.9, BCKO/BKOS. Suppose that a doubly-connected edge list of a connected subdivision is given. Give pseudocode for an algorithm that lists all faces with vertices that appear on the outer boundary.

(4). Write an intersection predicate, `TriCornerSegIntersectProp`, in C which takes five points,  $a, b, c, d, e$ , which define a triangle,  $\Delta abc$  and a segment  $de$ , determines if they "intersect properly" in the following sense: Does the segment  $de$  intersect the triangle  $\Delta abc$  in exactly one point  $p$  that is a corner of the triangle (one of  $a, b$ , or  $c$  – i.e., it should be that  $de \cap \Delta abc \in \{a, b, c\}$ ), and this point  $p$  lies interior to  $de$  (not at one of its endpoints)? (A triangle is considered to be the (closed) convex hull of its three vertices.) Make your code robust to the possibility of degeneracies: assume that the five points are all distinct, but that there may be collinearities among them (possibly even all 5 lie on a line, making the triangle  $\Delta abc$  degenerate).

Your predicate should start:

```
bool    TriCornerSegIntersectProp( tPointi a, tPointi b, tPointi c,
                                   tPointi d, tPointi e )
{
    [FILL IN HERE]
}
```

Make sure to give some brief explanation (possibly in the form of comment statements) to help the reader understand what your code is doing!

You may use the predicates from O'Rourke's book, `Area2`, `Left`, `LeftOn`, `Collinear`, `IntersectProp`, `Between`, and `Intersect` from the text, but try to think carefully about making your code complete (handling all cases) and efficient (avoiding duplication of tests).

*Note:* Even if you have never programmed in C before, this exercise should be fairly straightforward; try to read and understand the code fragments of section 1.5 of O'Rourke.

(5). Let  $S$  be a set of  $n$  line segments in general position in the plane (no three endpoints are collinear). Assume that there is at least one (proper) crossing point where two segments properly intersect. Let  $V$  denote the set of all (proper) crossing points. We wish to find the axis-aligned bounding box of  $V$ . Describe an efficient ( $O(n\log n)$ , ideally) method to do this. (Hint: Think about how to modify the Bentley-Ottmann sweep so that it can find the leftmost crossing point among a set of line segments.)