

# CS 266 Homework 2

Zachary DeStefano, PhD Student, 15247592

Due Date: April 17

### Problem 2.3

Change the code of Algorithm FINDINTERSECTIONS (and of the procedures that it calls) such that the working storage is  $O(n)$  instead of  $O(n+k)$ .

### Problem 2.11

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 discs, 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.

First, two circles will intersect in at most two points if they are not the exact same circle. Here is the proof: Any two circles can be translated and rotated so that one of the centers is the origin and the other center is on the  $x$ -axis. Thus assume the two circles have centers  $(0, 0)$  and  $(a, 0)$  and radii of  $r_1$  and  $r_2$ . We will assume distinct centers so that  $a \neq 0$ . The two circles are thus described by:

$$\begin{aligned}x^2 + y^2 &= r_1^2 \\(x - a)^2 + y^2 &= r_2^2\end{aligned}$$

Let  $R = r_1^2 - r_2^2$ . After subtracting the two equations we have

$$2ax - a^2 = R$$

We can turn this into

$$x = \frac{a^2 + R}{2a}$$

If  $x > r_1$  or  $x < -r_1$  then we know there is no intersection point. If  $x = r_1$  or  $x = -r_1$  then there is 1 intersection point. If  $-r_1 \leq x \leq r_1$ , then from the equation there is one matching  $x$  which means two matching  $(x, y)$  pairs, thus two intersection points. Since  $a \neq 0$  we do not have to worry about any more intersections.

### Problem 8.4

Let  $L$  be a set of  $n$  lines in the plane. Give an  $O(n \log n)$  time algorithm to compute an axis-parallel rectangle that contains all the vertices of  $A(L)$  in its interior.

If you take the dual of the  $n$  lines, you get  $n$  points.

An intersection occurs when

$$\begin{aligned}x &= \frac{b_2 - b_1}{m_2 - m_1} \\y &= \frac{b_2 m_1 - b_1 m_2}{m_2 - m_1}\end{aligned}$$

**Problem 8.14**

Let  $S$  be a set of  $n$  points in the plane. Give an  $O(n^2)$  time algorithm to find the line containing the maximum number of points in  $S$ .

Since a line through a pair of points in the primal plane becomes a vertex in the dual plane, we just have to compute the dual of the points and then compute which vertex has the most number of lines passing through it.

Here is the algorithm:

1. Take the dual of the  $n$  points
2. Use the arrangement algorithm to find the vertices.
3. Find which vertex contains the greatest number of lines.

Complexity analysis:

Taking the dual will take  $O(n)$  time.

Computing the arrangement takes  $O(n^2)$  time.

Going through the vertices will take  $O(n^2)$  time