

COS 226 Programming Assignment

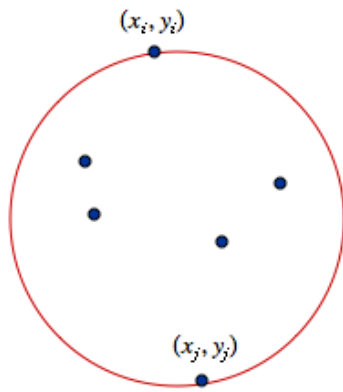
Smallest Enclosing Circle

DRAFT

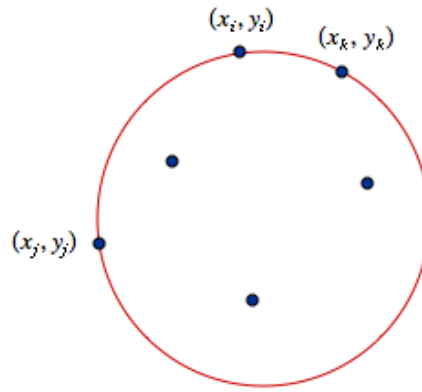
Given N points in the plane, find the circle of minimum radius that contains all of the points.

Applications. The problem has numerous applications. For example, consider the placement of a new school to service a number of students. If we model the students as points in the plane, the smallest enclosing circle specifies where to build the school: placing the school at the circle's center minimizes the distance between the school and the furthest student. Alternatively, if we view the points as military targets, the minimum enclosing circle specifies where to drop a bomb to destroy the targets, and its radius tell us how much explosive is required.

Structure. The smallest enclosing circle is uniquely defined. Moreover, it is characterized by two or three of the points, which lie on the boundary of the circle.



A smallest enclosing circle defined by 2 points.



A smallest enclosing circle defined by 3 points.

Specifically, the smallest enclosing circle is one of the following:

- A circle whose boundary contains 2 of the N points and whose diameter connects the two points. If the two points are (x_i, y_i) and (x_j, y_j) , then the circle has center and radius:

$$x_c = \frac{x_i + x_j}{2}, \quad y_c = \frac{y_i + y_j}{2}, \quad r_c^2 = \frac{(x_i - x_j)^2 + (y_i - y_j)^2}{4}$$

- A circle whose boundary contains 3 of the N points. If the three points are (x_i, y_i) , (x_j, y_j) , and (x_k, y_k) , then the circle has center and radius:

$$x_c = \frac{d}{2a}, \quad y_c = \frac{-e}{2a}, \quad r_c^2 = \frac{d^2 + e^2}{4a^2} + \frac{f}{a}, \quad \text{where}$$

$$a = \begin{vmatrix} x_i & y_i & 1 \\ x_j & y_j & 1 \\ x_k & y_k & 1 \end{vmatrix}, \quad d = \begin{vmatrix} x_i^2 + y_i^2 & y_i & 1 \\ x_j^2 + y_j^2 & y_j & 1 \\ x_k^2 + y_k^2 & y_k & 1 \end{vmatrix}, \quad e = \begin{vmatrix} x_i^2 + y_i^2 & x_i & 1 \\ x_j^2 + y_j^2 & x_j & 1 \\ x_k^2 + y_k^2 & x_k & 1 \end{vmatrix}, \quad f = \begin{vmatrix} x_i^2 + y_i^2 & x_i & y_i \\ x_j^2 + y_j^2 & x_j & y_j \\ x_k^2 + y_k^2 & x_k & y_k \end{vmatrix}$$

Recall, the *determinant* of a 3-by-3 matrix is defined as follows:

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11}(a_{33}a_{22} - a_{32}a_{23}) - a_{12}(a_{33}a_{21} - a_{31}a_{23}) + a_{13}(a_{32}a_{21} - a_{31}a_{22})$$

Brute force. Write a program `Brute.java` that examines all pairs and all triples of points, checks which of the corresponding circles encloses all N points, and outputs the smallest such circle.

An efficient algorithm.

Input format. The data file consists of an integer N , followed by N pairs of real numbers (x, y) . The points will all lie in the unit circle: $(x^2 + y^2 \leq 1)$.

```
% more input5.txt          % more input8.txt
```

Output format. Print the circle on standard output and draw the points and the circle on standard draw. Scale the coordinate system so that coordinates between -1 and +1 fit in the graphics window.

```
% java Brute < input5.txt
output
```

```
% java Fast < input8.txt
output
```

Analysis. Estimate (using tilde notation) the average-case running time (in seconds) of your two programs as a function of the number of points N (where the points are generated uniformly in the unit circle). Provide empirical evidence to justify your hypotheses.

What is the worst-case running time of your two program as a function of N ?

Deliverables. Submit the files: `readme.txt`, `Brute.java`, `Fast.java`, `Point.java`, and `Circle.java`. Also submit any other auxiliary files, if any, that your program needs (excluding `std*`).

*This assignment was developed by Kevin Wayne.
Copyright © 2008.*