

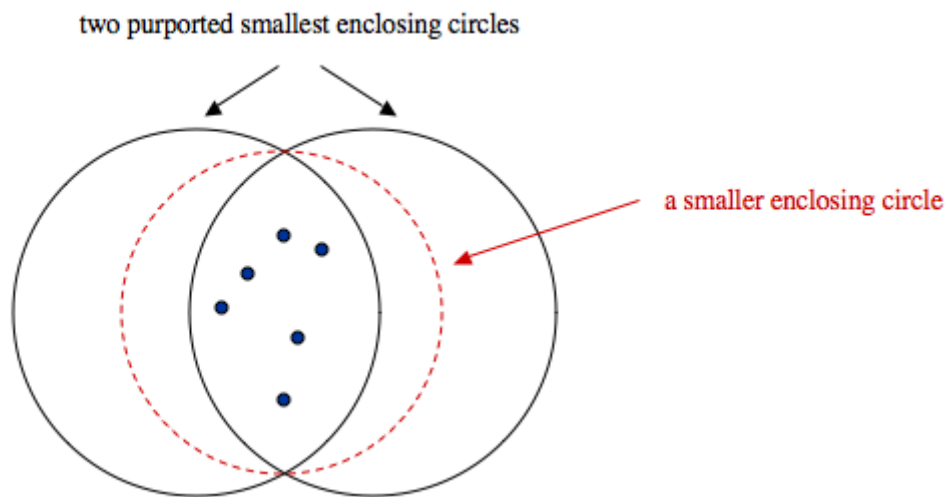
COS 226 Programming Assignment Checklist: Smallest Enclosing Circle

Demo

The following applet (written by Xiumin Diao) draws the smallest enclosing circle. Click to add points; drag a point to move it.

Frequently Asked Questions

Why is there a unique smallest enclosing circle? Consider two smallest enclosing circles C_1 and C_2 of radius r and centers c_1 and c_2 . Consider the circle C centered at $1/2 (c_1 + c_2)$ and radius $\sqrt{r^2 - a^2}$, where a is one-half the distance between c_1 and c_2 . C encloses all of the points since it contains $C_1 \cap C_2$.



Floating point precision.

In the formula for the center and radius of a circle defined by 3 points in the boundary, what happens when a is zero? Observe that the formula for a is precisely the formula for computing the ccw of three points. When a is zero, the three points are collinear, so they cannot be on the boundary of the same circle (assuming they are distinct).

What should my program do if the input consists of one (or zero) points? In the first case, output a circle centered at the point of radius 0; in the second case report that the input has no points.

What should my program do if a point appears more than once in the input? Handle it correctly.

Are the two or three boundary points that define the circle on the convex hull? Yes. Feel free to exploit this fact to optimize your solution.

What's the worst-case running time?

My program results in a stack overflow exception for large N . Is this OK? A perfect solution should avoid an excessive stack size, e.g., by converting the recursive function into a non-recursive one using an explicit stack.

Testing and Submitting

Input. Here are some [sample input files](#). Thanks to Jesse Levinson '05 for the remarkable input file `rs1423.txt`. To estimate the running time as a function of N , you may use the program [Generator.java](#). It takes a command-line argument N and outputs a random instance of N points.

Reference solutions. Some of the input files have associated `.png` files that contain the desired output. You can use these for checking your work.

Readme. Use the [following readme file template](#) and answer all questions. This includes analyzing the two algorithms both empirically and analytically.

Possible Progress Steps

These are purely suggestions for how you might make progress. You do not have to follow these steps.

1. **Getting started.** Download the directory [circle](#) to your system. It contains a number of sample input files.
2. **Brute force.** Before writing any code, think carefully about the point and circle data type operations that you will need to support.
3. **Faster algorithm.**

Enrichment

Link to papers.

wayne@cs.princeton.edu