Data och Informationsteknik / Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
Erland Holmström Göteborg 14 Sep 2017

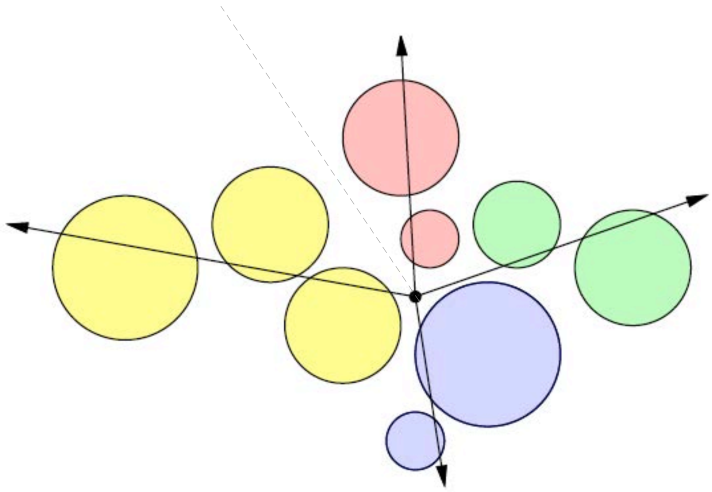
In this course it's very important that you motivate what you do (justify and explain both what you are doing and why you are doing it) so it's simple to follow your line of argument.

- You should hand in one pdf file per problem. Name the files `ass2.1.pdf` and `ass2.2.pdf`. Write your name in each file. You don't need to compress. And note that Fire has one entry per problem, not one per assignment. This makes it easier to divide the problems between the correctors.
- **Be careful to follow the "How to present algorithms" given in lectures and on the web. Step 1-4 and perhaps some on 5 is expected if not stated otherwise.**

Assignment 2 problem 1, Algorithms.

Covers: "In order to get what you want, just start grabbing what looks best."

Suppose you are standing in a field surrounded by several large balloons. You want to use your brand new Acme Brand Zap-O-MaticTM to pop all the balloons, without moving from your current location. The Zap-O-MaticTM shoots a high-powered laser beam, which pops all the balloons it hits. Since each shot requires enough energy to power a small country for a year, you want to fire as few shots as possible.



Nine balloons popped by 4 shots of the Zap-O-Matic™

The minimum zap problem can be stated more formally as follows. Given a set C of n circles in the plane, each specified by its radius and the (x, y) coordinates of its centre, compute the minimum number of rays from the origin that intersect every circle in C. Your goal is to find an efficient algorithm for this problem.

a) Suppose it is possible to shoot a ray that does not intersect any balloons (for example the dotted line above). Describe and analyse a greedy algorithm that solves the minimum zap problem in this special case.
Any solution/complexity will award some points (amount depends on complexity and how nice the solution is but max half the points) but for full points you need to give a greedy algorithm with $O(n \log n)$ time. Tip: use stay ahead for the proof.

b) Describe and analyse a greedy algorithm (without the special case in a) whose output is within 1 of optimal. That is, if m is the minimum number of rays required to hit every balloon, then your greedy algorithm must output either m or $m+1$. (Of course, you must prove this fact.) If your solution is similar to the one in a) then only describe the difference.

Assume you have a subroutine `intersects(r, c)` that determines whether a ray r intersects a circle c in $O(1)$ time. This subroutine is not difficult to write, but it's not the interesting part of the problem.

Assignment 2 problem 2.

A large ship is to be loaded with cargo. The cargo is packed in containers, and all containers are the same size. Different containers may have different weights though. Let w_i be the weight of the i'th container, $1 \le i \le n$. The cargo capacity of the ship is c. We wish to load the ship *with the maximum number of containers*. This problem can be formulated as an optimization problem in the following way: Let x_i be a variable whose value can be either 0 or 1. If we set x_i to 0, then container i is not to be loaded. If x_i is 1, then the container is to be loaded. We wish to assign values to the x_is that satisfy $\sum_{i=1}^{n} w_i x_i \leqslant c$ and $x_i \in \{0,1\}, 1 \leqslant i \leqslant n$.

The optimization function is $max \sum_{i=1}^{n} x_i$

(The algorithm for this problem is trivial so expect most of the points to be in the proof. Try a exchange arguments proof.)

(preliminary distribution of points in the exam was: idea: 4, proof 10, pseudocode: 2, complexity 2 18p)