

# Scenario Week 4

## (comp203p)

Ilya Sergey



`scenario@cs.ucl.ac.uk`

22-26 February 2016

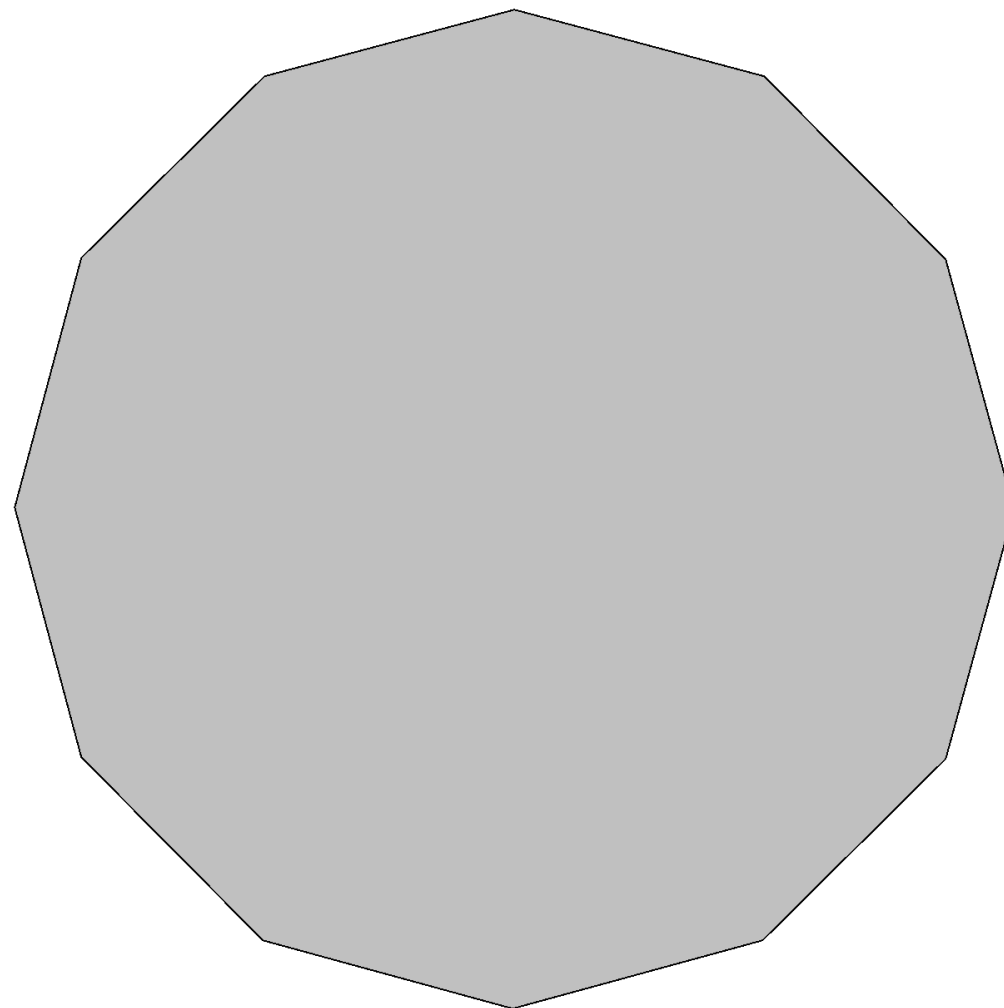






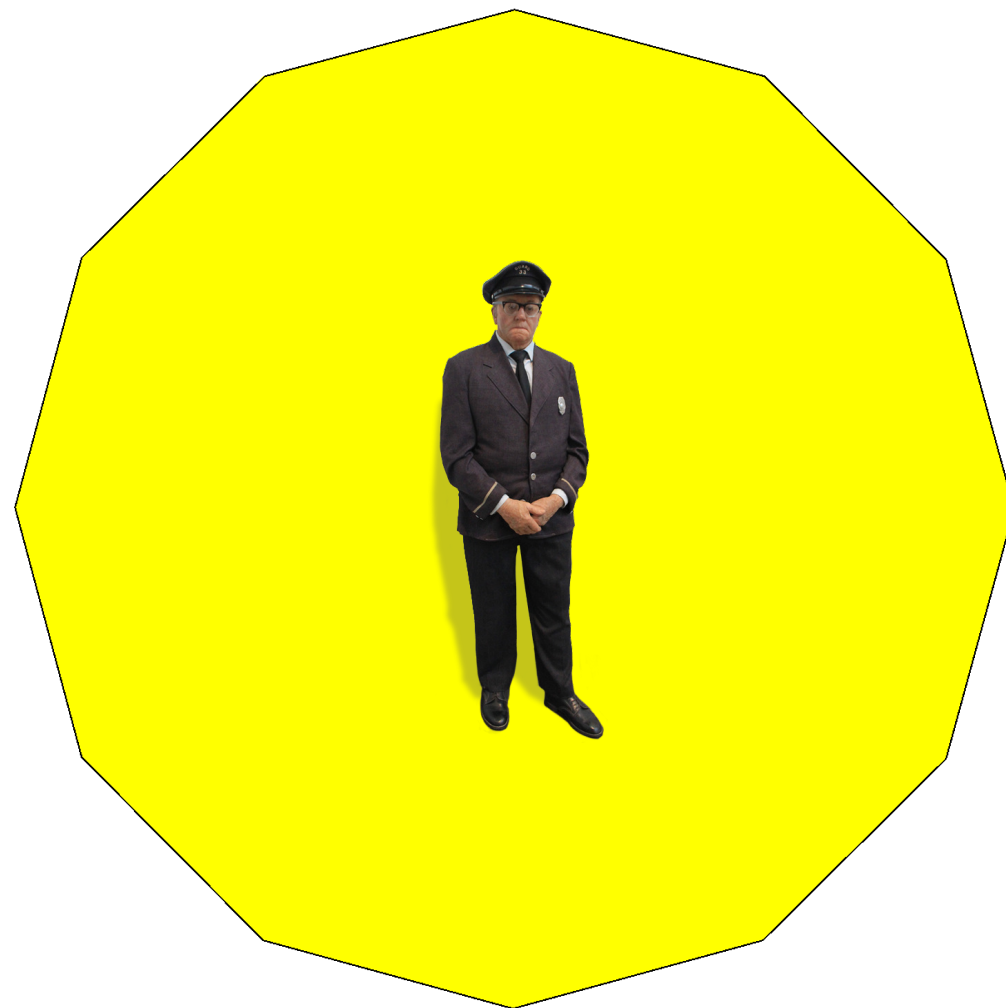
# How many guards do we *really* need?

The answer depends on the shape of the gallery.



# How many guards do we *really* need?

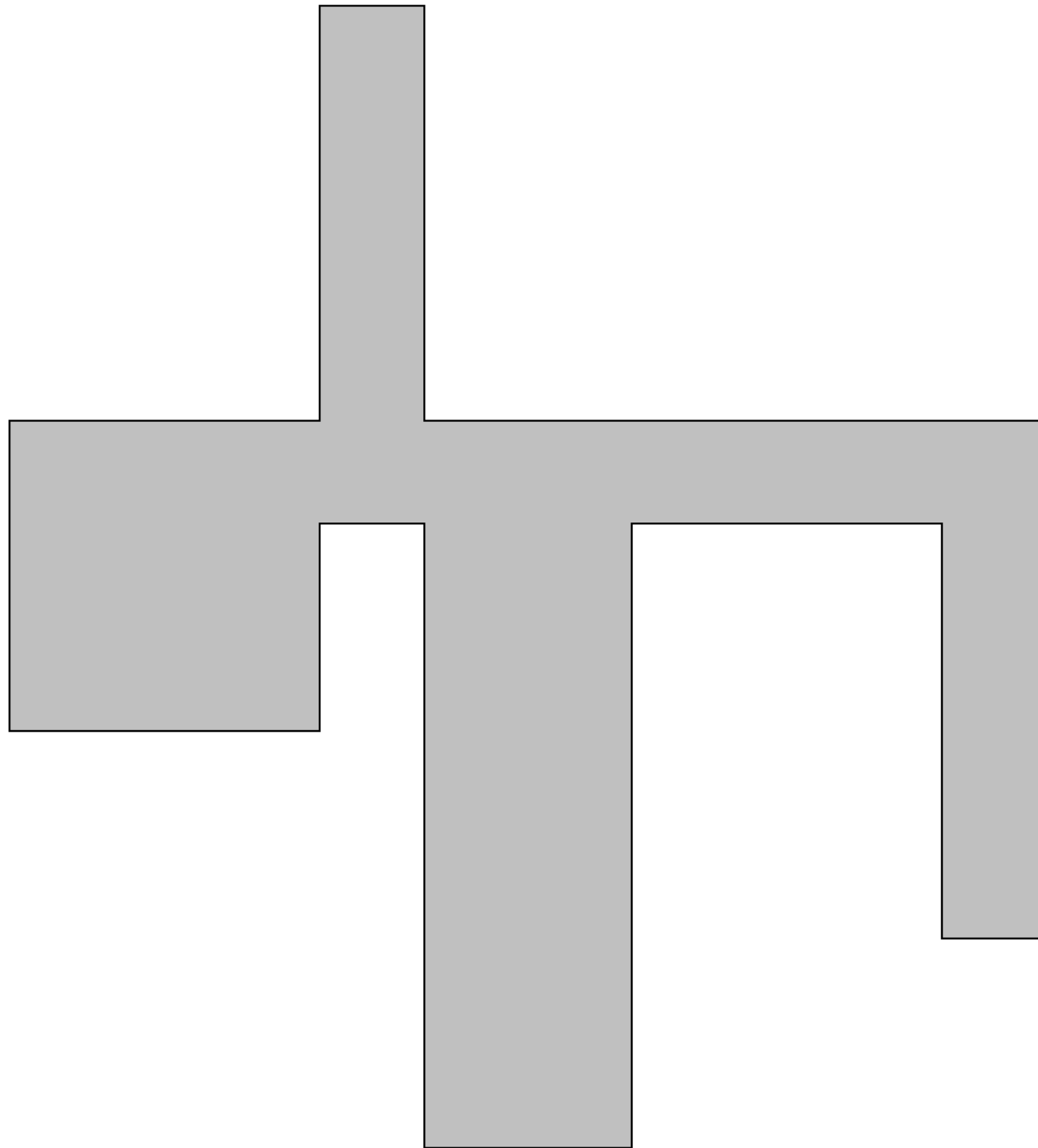
The answer depends on the shape of the gallery.



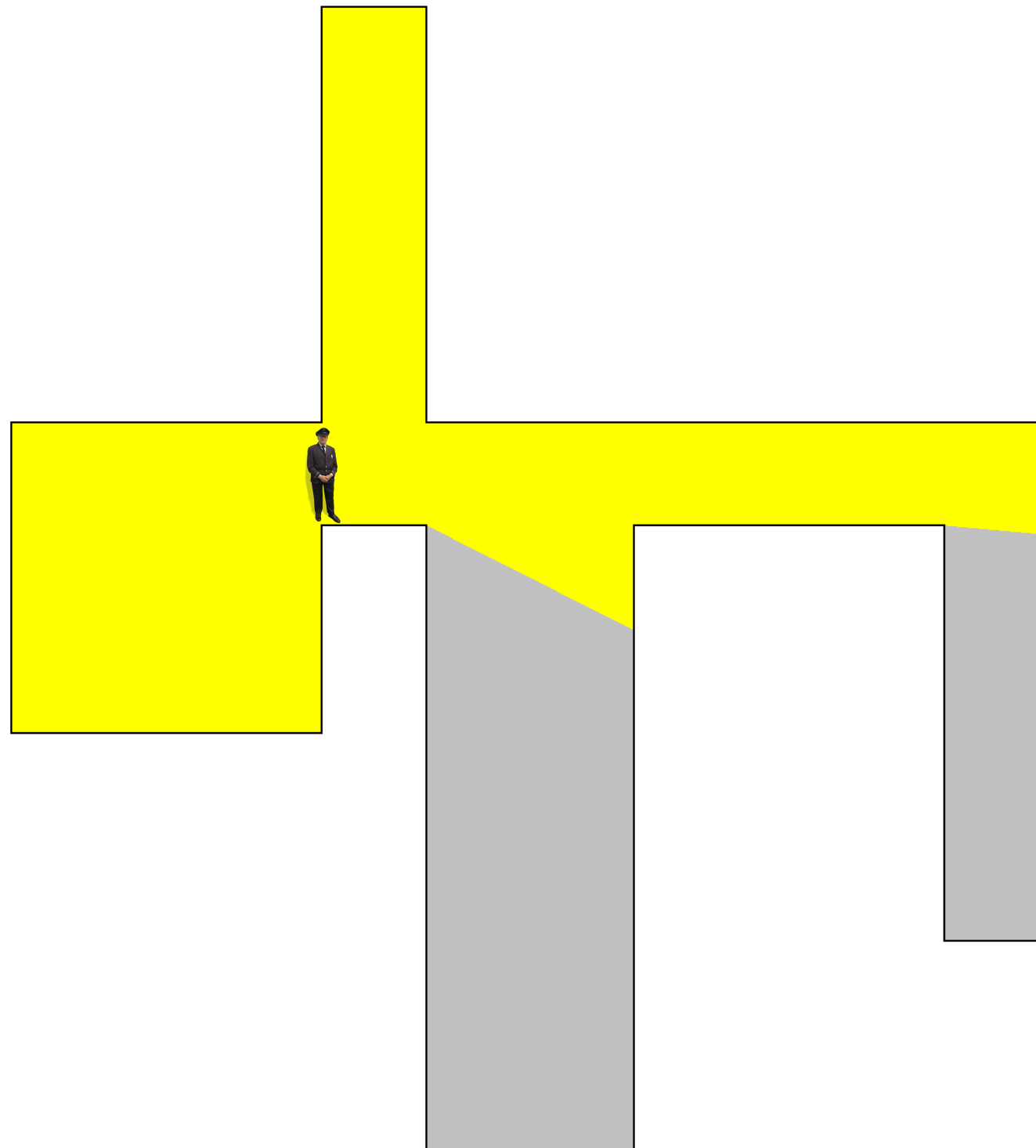
Here just 1 guard is okay.



How many guards do we *really* need?

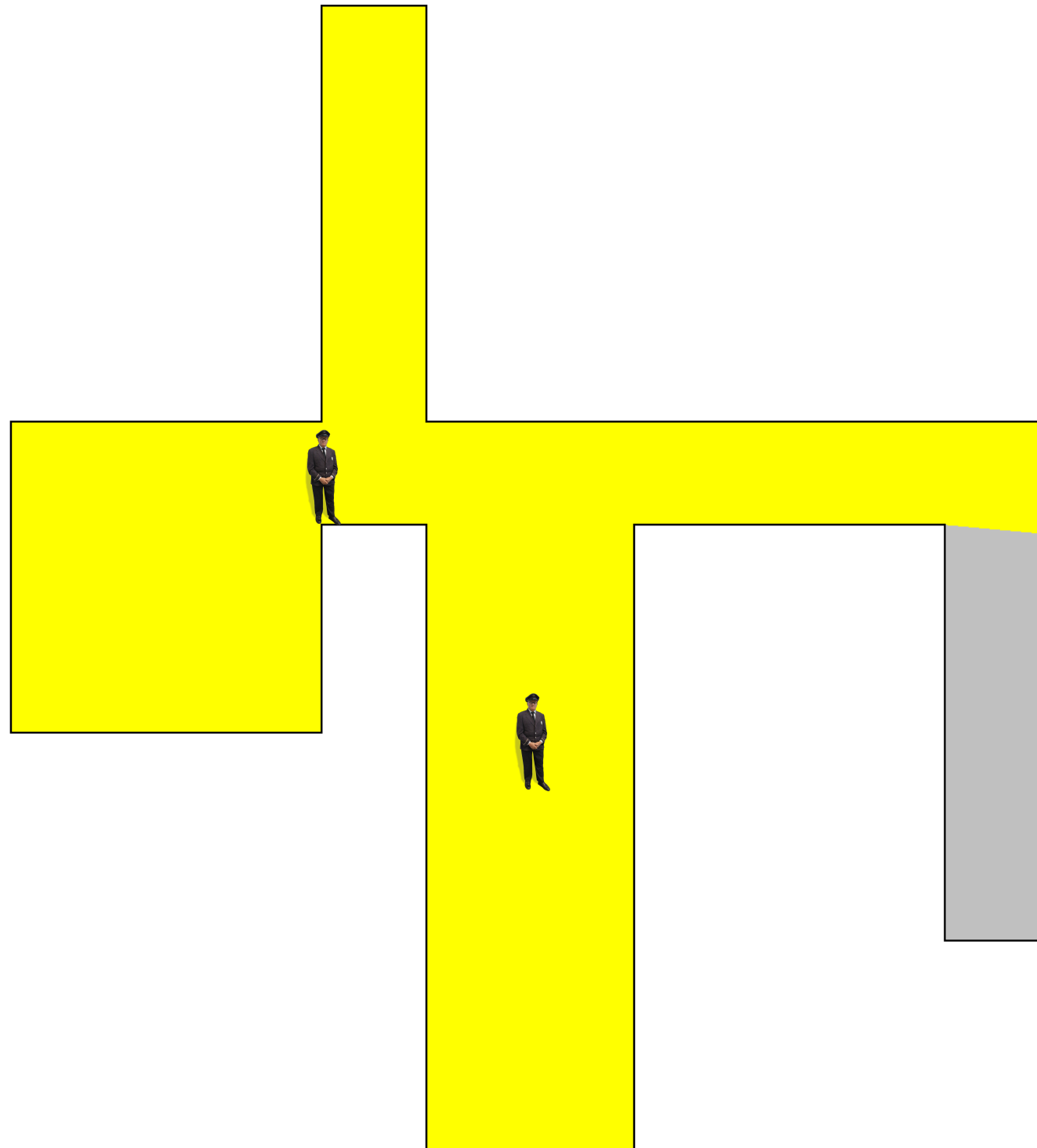


How many guards do we *really* need?

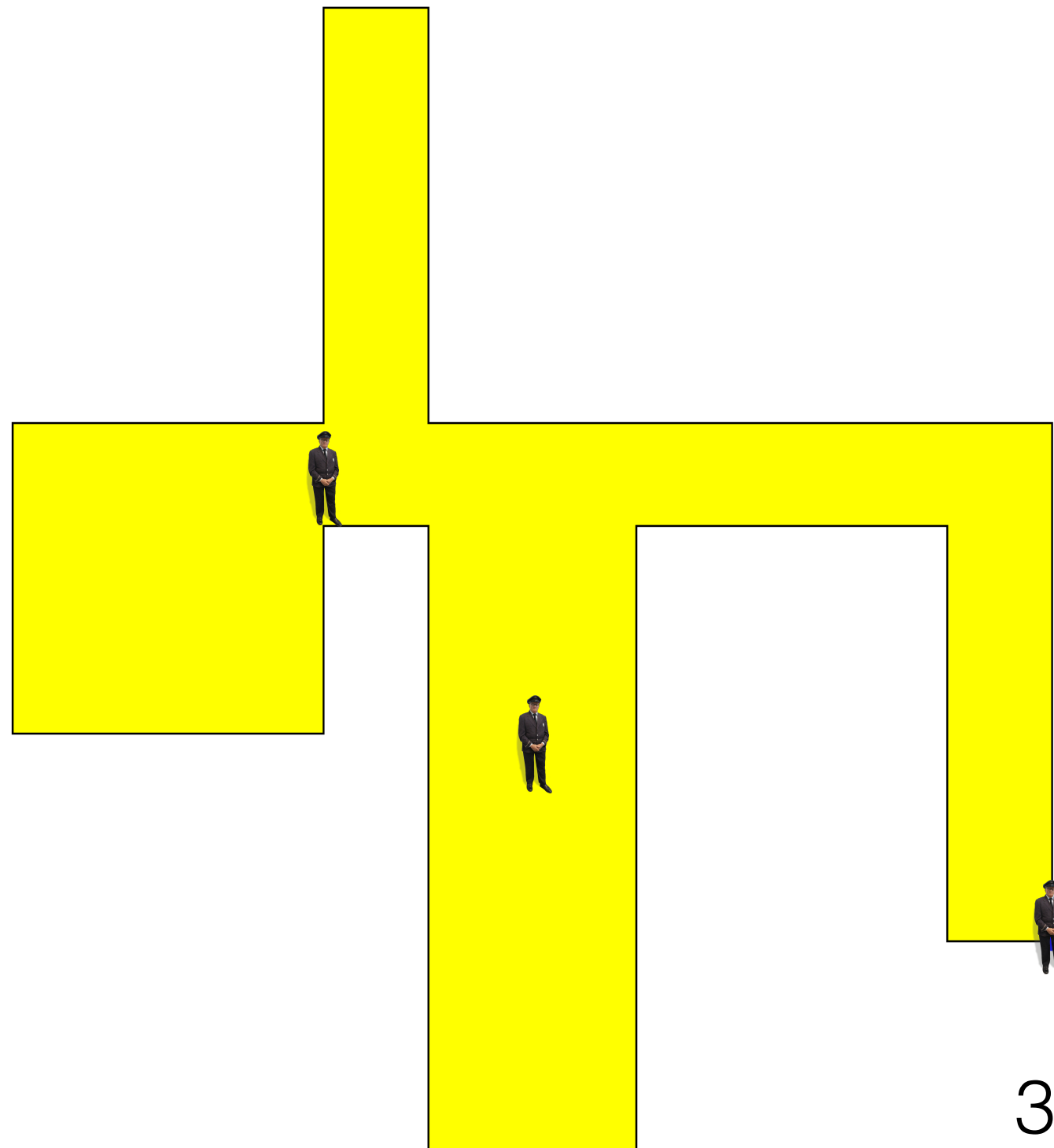




How many guards do we *really* need?



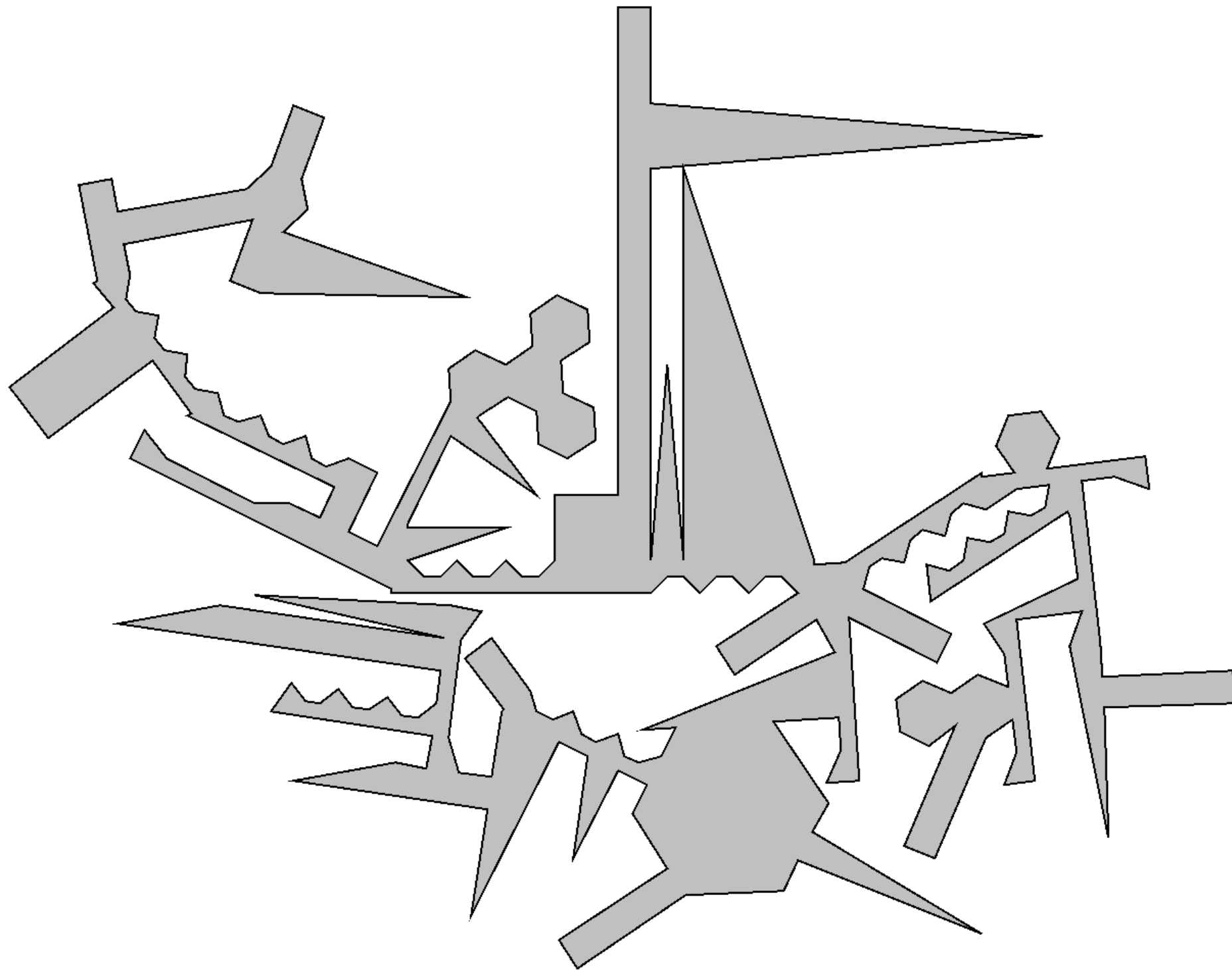
# How many guards do we *really* need?



3 guards will do.



How many guards do we *really* need?



# Art Gallery Problem

For a given gallery (polygon),  
find the *minimal* set of guards' positions,  
so together the guards can “see” the *whole* interior.

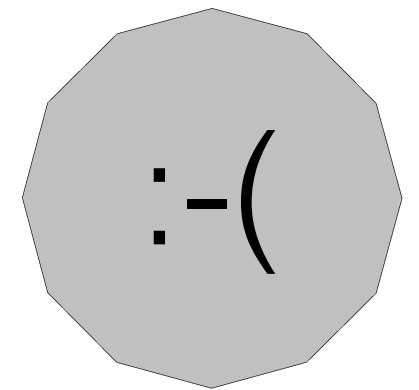
- *Complexity-wise, harder than*
  - SAT
  - Travelling salesman
  - Hamiltonian paths
  - Knapsack problem

NP-hard



# Cheap-and-cheerful “almost” solutions

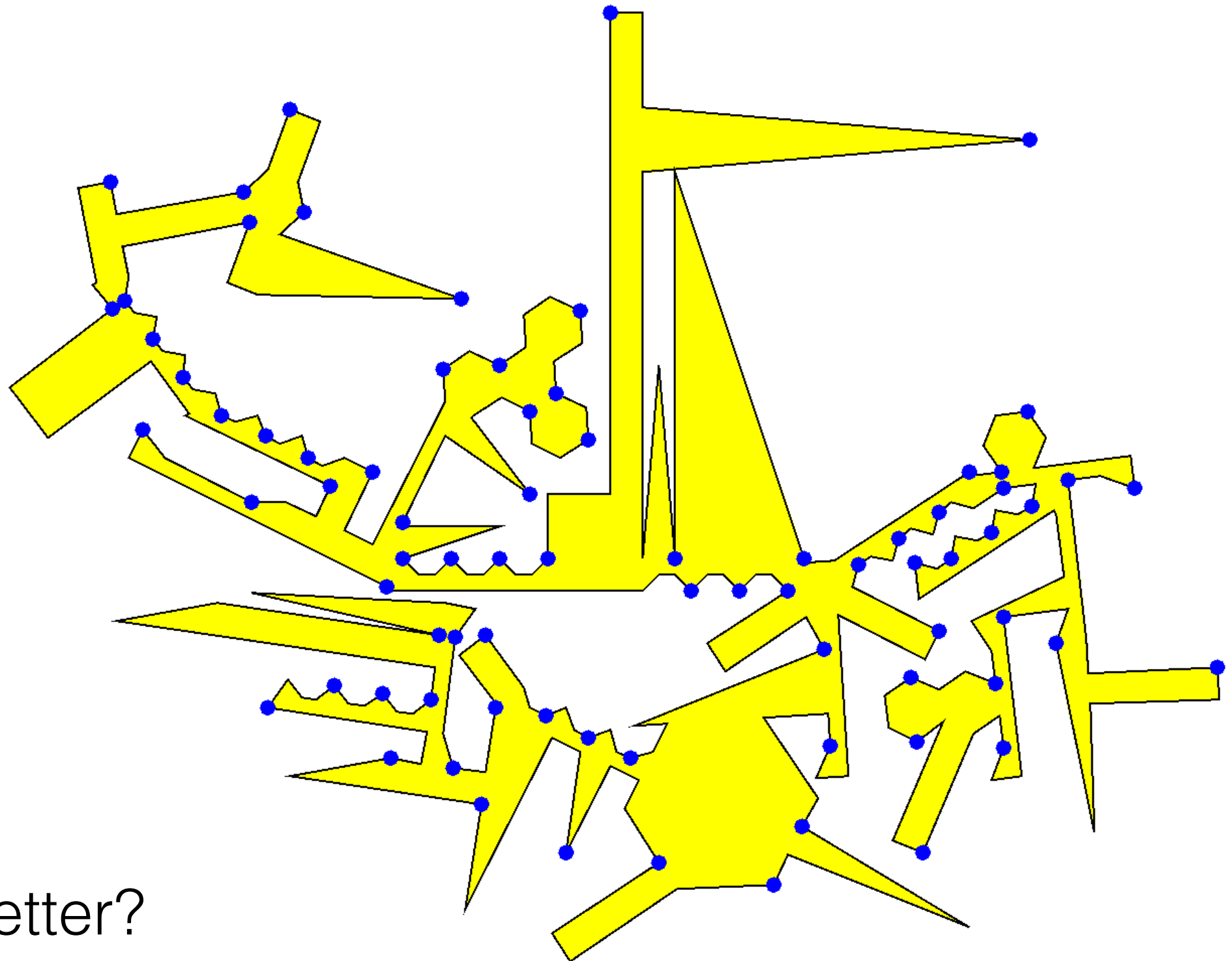
- Putting guard in *each* vertex
  - ▶  $n$  guards for a polygon with  $n$  vertices
- Václav Chvátal’s solution (1975)
  - ▶ based on *triangulation*,  $\lfloor n/3 \rfloor$  guards;
  - ▶ **Chvátal’s theorem**: this number is *always sufficient* and is in some cases *necessary*.



# Chvátal's solution in practice

- 246 vertices
- 79 guards

Can we do better?





# Scenario Week 4

(comp203p)

## *Art Gallery Competition*

`scenario@cs.ucl.ac.uk`

22-26 February 2016

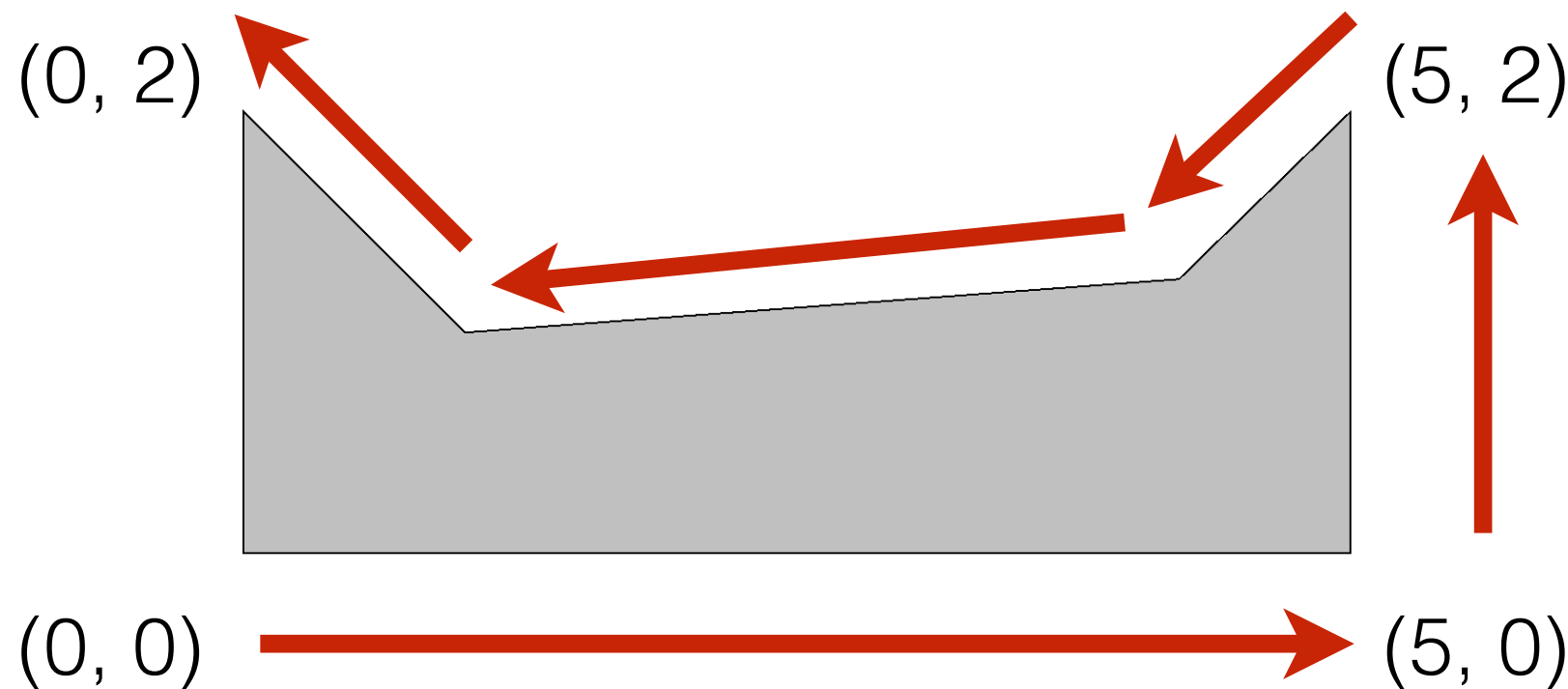
# Part I: Computing “good enough” set of guards

- **30** galleries of different shapes;
  - File with galleries: **guards.pol** (see Moodle page);
  - sizes of problems: small ( $< 10$ ) to large ( $\sim 300$ );
- Compute a *complete* set of guards for *each one* of them;
- *Baseline* — Chvátal's boundary (cannot get worse than that);
- Grading: **30 points**, *one per gallery*, for *any* solution, which is not worse than the baseline.

# Encoding of the problems (Part I)

guards.pol

1:	(0, 0), (2, 0), (2, 1), (1, 1), (1, 3), (0, 3)
2:	(0, 0), (5, 0), (5, 2), (4.2312351, 1.234), (1, 1), (0, 2)



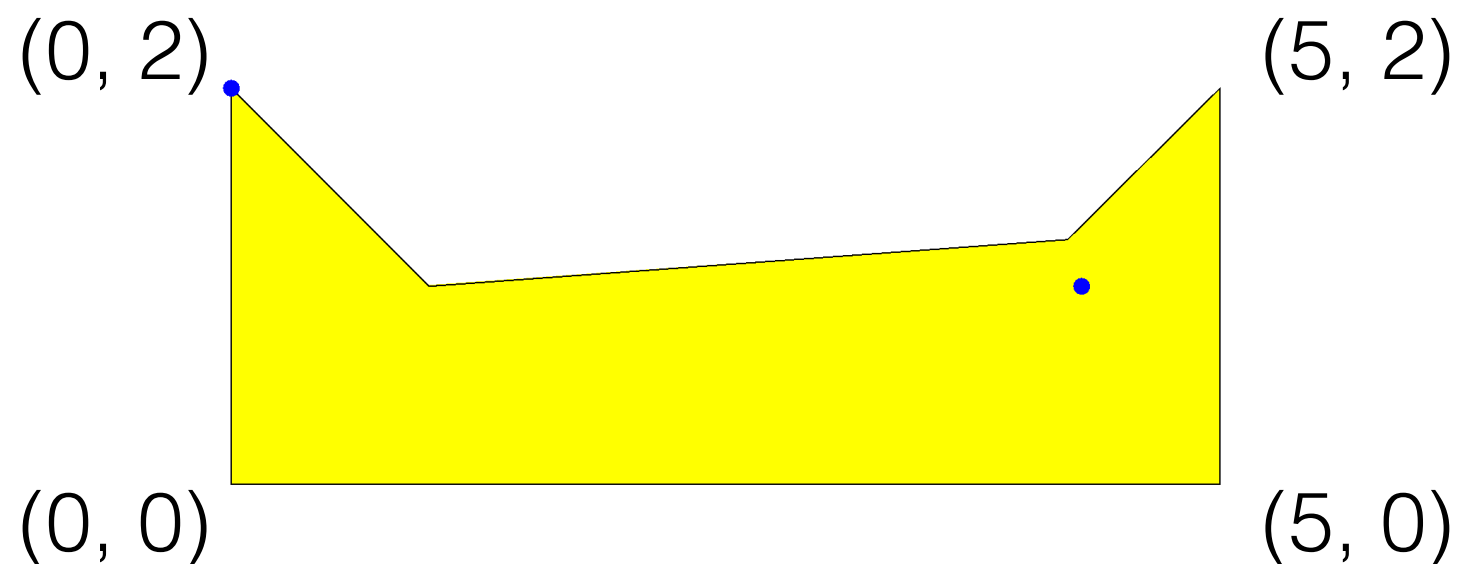
- Polygon is “on the left”
- No holes inside

# Encoding your solutions (Part I)

Solution file:

team name  
team's password  
per-polygon guards

```
tiger
lt671vecrskq
2: (0, 2), (4.3, 1)
1: (0.2, 2.5), (2, 0.5)
```





# Checking and submitting solutions

- **Warning:** *double-precision floating-point* arithmetic
  - all equalities are up to  $\epsilon = 0.000,000,000,1$
- Details on acceptance criteria are in the [specification](#) (on Moodle)
- Submit your solutions here (under Part I):

<http://artgallery.cs.ucl.ac.uk>

Solutions are accepted until **14:00 GMT 26 Feb 2016**

## Part 2: Checking a (flawed) set of guards

- **20** galleries of different shapes *with* sets of guards;
  - File with problems: **check.pol** (see Moodle page);
  - sizes of problems: small ( $< 10$ ) to gigantic ( $\sim 500$ );
- Find a *refutation* (a point within a polygon, not visible from the given guards) for *each* problem in the set;
- *Any* refutation will do.
- Grading: **20 points**, *one per problem/refutation*.

# Encoding of the problems (Part 2)

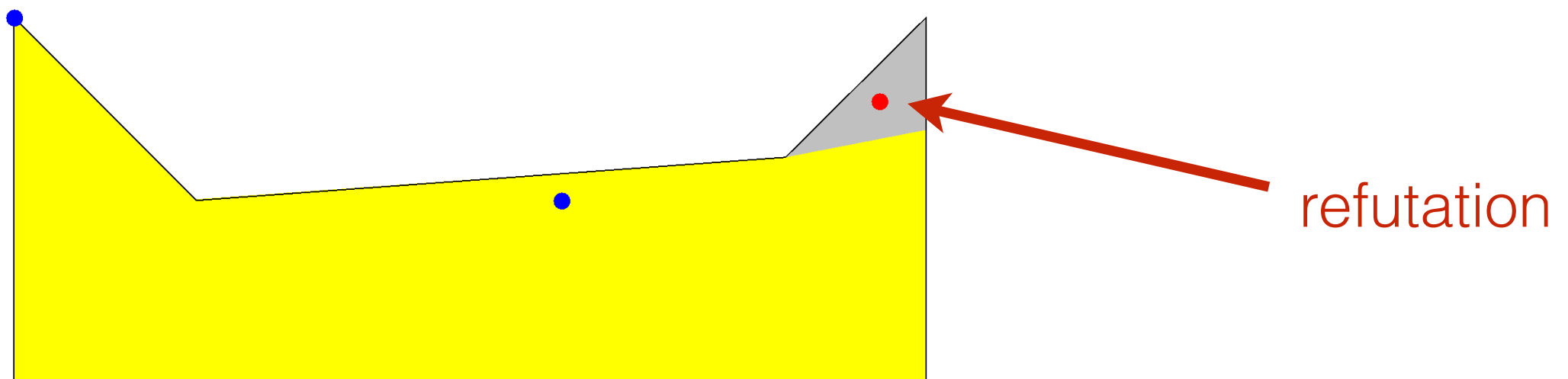
File with problems

check.pol

1:	(0, 0), (2, 0), (2, 1), (1, 1), (1, 3), (0, 3); (0, 3), (1, 2)
2:	(0, 0), (5, 0), (5, 2), (4.2312351, 1.234), (1, 1), (0, 2); (0, 2), (3, 1)

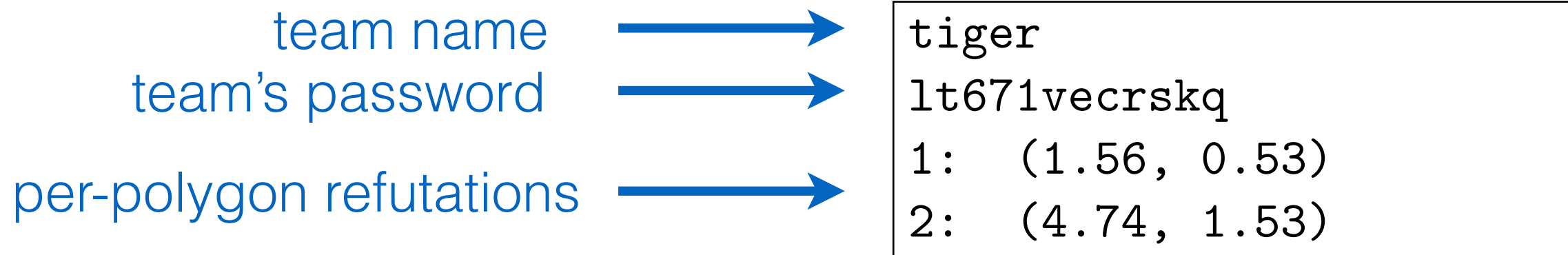
polygon vertices

guards



# Encoding your solutions (Part 2)

Solution file:



- Submit your solutions here (under Part 2):

<http://artgallery.cs.ucl.ac.uk>

Solutions are accepted until **14:00 GMT 26 Feb 2016**

# Part 3: Visualisation

- Implement a visualiser for galleries, guards and visibility:
  - drawing galleries;
  - drawing visibility areas from specific guards;
  - drawing refutations for incomplete guard sets.
- Grading: **15 points**
- Assessed by the organisers from **14:00 till 17:00, 26 Feb 16**
  - **book a slot for your team!**



# Part 4: Implementation report

- Describe your implementation experience
  - language, algorithms, *etc.*
  - details in the specification (see Moodle)
- Grading: ***15 points***
- Submit electronically by **17:00, 26 Feb 2016** (one per team)

# Part 5: The Competition!

- Compete with other teams for the *best* solutions in Part 1.
- Teams with *all* accepted solutions ranked amongst each other first.
- Check the score table <http://artgallery.cs.ucl.ac.uk> at for details
- Grading: up to **20 points**.

Rank	Score
1	20
2-3	15
4-5	10
6-7	5
$\geq 8$	0

# Overall grading

Task	Max grade
Computing “good enough” guard set	30
Checking a flawed guard set	20
Visualisation of the solutions	15
Implementation report	15
The Competition	20

# This week schedule

	Monday, 22 Feb	Tuesday, 23 Feb	Wednesday, 24 Feb	Thursday, 25 Feb	Friday, 26 Feb
10:00-11:00	ULU Malet Suite (Introductory lecture)	Roberts 421	Bedford Way LG04	Roberts 106	Roberts 421
11:00-13:00		Christopher Ingold XLG2 Auditorium	Chadwick B05 LT	Medawar G01 Lankester LT	Cruciform B404 - LT2
13:00-14:00	Lunch	Lunch	Birkbeck Clore Management Centre B01	Lunch	Lunch
14:00-16:00	Cruciform B404 - LT2	Cruciform B304 - LT1		Medawar G01 Lankester LT	Birkbeck Malet Street B36
16:00-18:00	Roberts 106	Cruciform B304 - LT1		Medawar G01 Lankester LT	Roberts G06 Sir Ambrose Fleming LT (Concluding lecture at 17: 00)

Helpdesk (green) = Time and locations where staff and/or TAs will be present so you could ask questions.

Lectures (blue) = Introductory and concluding lectures

# Good luck!

