

Distributed Algorithms 2022



Welcome!

- You should have already done this:
 - register in **Sisu**
 - read instructions in MyCourses
 - join our **Zulip workspace**
 - watch two pre-recorded videos
 - solve this week's quiz

This week: extra time to solve the quiz until midnight today!

Our weekly routine

- Mon: prerecorded videos
- **Tue:** quiz (noon), lecture (12:15pm)
- Wed: 1 exercise (midnight)
- Thu: exercise session (10:15am)
- Fri: 2 exercises (midnight)

Workload: 10–11 h/week



- One quiz per week, in the textbook
- Solve by *Tuesday* at noon
- Submit your answer in **MyCourses**
 - type the answer in the web form
- The answers should be very short
 - just give the answer, nothing else!
 - no proofs, no explanations!

Exercises

- 5+ exercises per week, in the textbook
- Solve 1 by *Wednesday*, 2 more by *Friday*
- Submit your answers in **MyCourses**
 - submit the answer as an easy-to-read PDF file
- The answers need to be complete
 - full details, complete proofs
 - e.g. why does your algorithm work correctly?

Challenging exercises

- In the textbook, marked with a star \bigstar
- Solve *at any point* during the course
- Email your answers to the lecturer • with full details, as an easy-to-read PDF file
- Again, the answers need to be complete



• To pass the course:

• you need to pass both *exams*

•For a good grade:

- you need to *solve exercises*
- quiz + exercises = max 96 points in total
- challenging exercises = 4 extra points each
- 80 points = grade 5/5

Learning objectives

- Understand models of distributed computing
- Design and analyze efficient distributed algorithms
- Prove impossibility results
- Use standard graph-theoretic concepts

Practiced in exercises Tested in midterm exams

This is a theory course

100% mathematics

- definitions
- theorems
- proofs ...

0% practice

- programming
- hardware
- protocols ...

Expected: basic knowledge of university-level mathematics

Example: what is a mathematical proof

Course practicalities

- Traditional on-campus course
 - on-campus lectures
 - on-campus exercise sessions
 - on-campus exams
- Primary tool for communication: **Zulip**
- Course material, submitting solutions:
 MyCourses

This week's content...

Video 1a: introduction



Quick question to all

What was the **key new thing** to you in the first video?

Or were you already familiar with the distributed perspective?

Video 1b: coloring



Slow color reduction

- Algorithm idea:
 - all nodes with the **largest color** are active
 - active nodes pick the *smallest color that is not used by their neighbors*



Group work 1

- Consider a simpler algorithm idea:
 all nodes pick the smallest color that is not used by their neighbors
- What would go wrong?
 - construct an example in which this algorithm fails!

Video 1b: coloring fast

Fast color reduction

- Algorithm idea:
 - find the first bit that differs in successor
 - index *i*, bit value *b*
 - new color is (*i*, *b*)

Group work 2

- Algorithm idea:
 - find the first bit that differs in successor
 - index *i*, bit value *b*
 - new color is (*i*, *b*)
- What would go wrong if the new color was just b?
 - construct an example in which it fails!

Group work 3

- Algorithm idea:
 - find the first bit that differs in successor
 - index *i*, bit value *b*
 - new color is (*i*, *b*)
- What would go wrong if the new color was just i?
 - construct an example in which it fails!

Coming next

- Week 2: graph theory
- Weeks 3–6: models of distributed computing • examples of efficient distributed algorithms
- Weeks 7–11: proving impossibility results
- Week 12: conclusions, recap