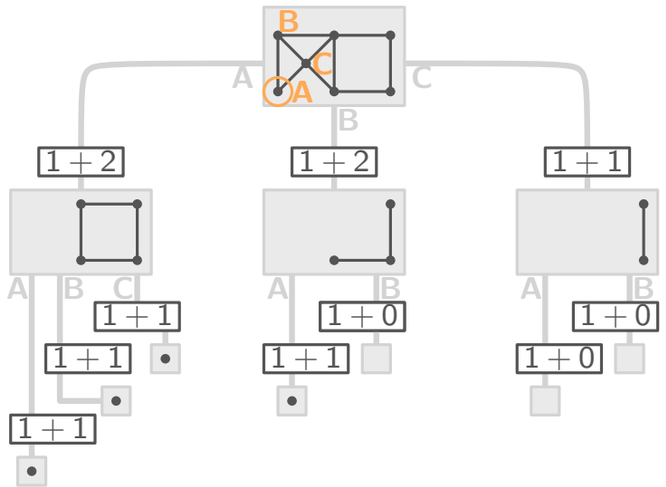
Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

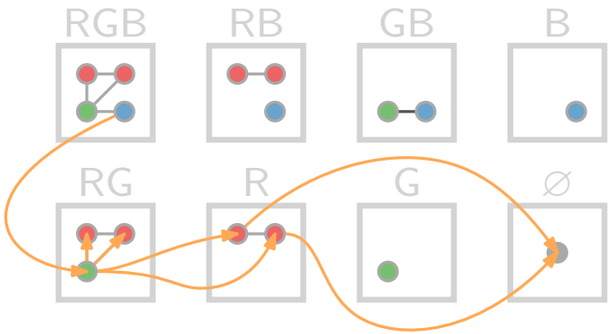
Advanced Algorithms

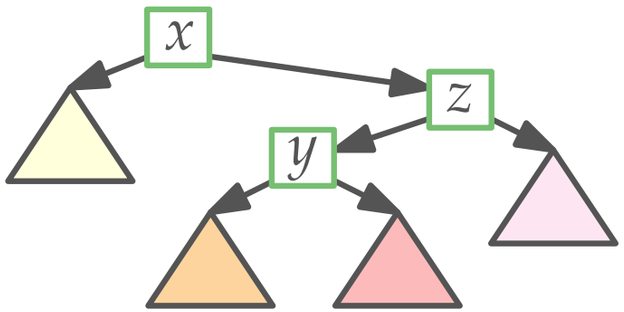
Introduction

Topics, Course Details, Organizational

Johannes Zink · WS23/24







Advanced Algorithms

The goal of this course is to offer an overview of advanced algorithmic topics.

Advanced Algorithms

The goal of this course is to offer an overview of advanced algorithmic topics.

You have already learned a lot about algorithms

Advanced Algorithms

The goal of this course is to offer an overview of advanced algorithmic topics.

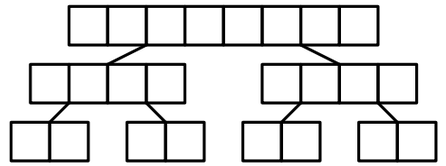
You have already learned a lot about algorithms,
but there is much more left...

Advanced Algorithms

The goal of this course is to offer an overview of advanced algorithmic topics.

You have already learned a lot about algorithms,
but there is much more left...

- **Types:** incremental, recursive, D&C, greedy, numerical

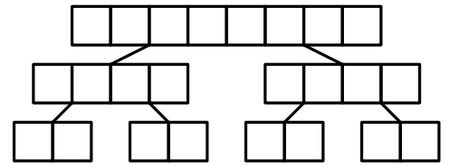


Advanced Algorithms

The goal of this course is to offer an overview of advanced algorithmic topics.

You have already learned a lot about algorithms,
but there is much more left...

- **Types:** incremental, recursive, D&C, greedy, numerical, exact, approx., randomized, parallel, distributed, ...

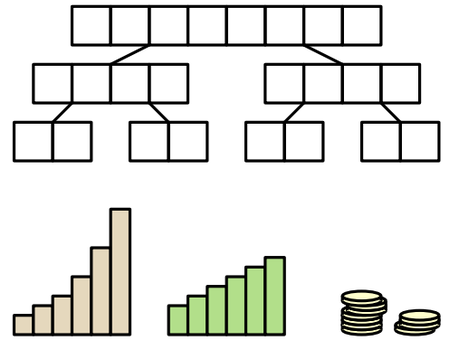


Advanced Algorithms

The goal of this course is to offer an overview of advanced algorithmic topics.

You have already learned a lot about algorithms,
but there is much more left...

- **Types:** incremental, recursive, D&C, greedy, numerical, exact, approx., randomized, parallel, distributed, ...
- **Analysis:** correctness, runtime, space usage, amortized

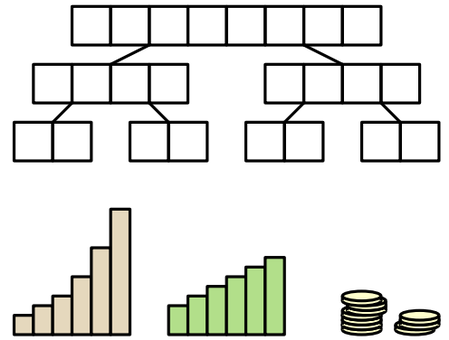


Advanced Algorithms

The goal of this course is to offer an overview of advanced algorithmic topics.

You have already learned a lot about algorithms,
but there is much more left...

- **Types:** incremental, recursive, D&C, greedy, numerical, exact, approx., randomized, parallel, distributed, ...
- **Analysis:** correctness, runtime, space usage, amortized, expected, optimality, benchmarking, ...

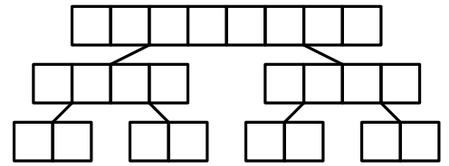


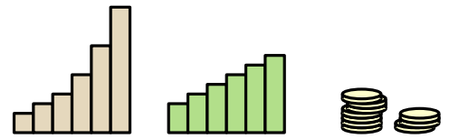
Advanced Algorithms

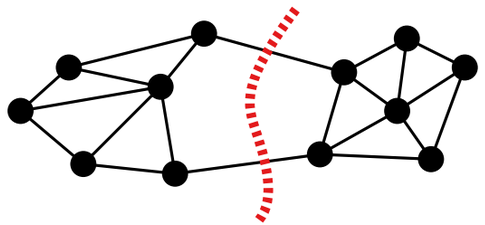
The goal of this course is to offer an overview of advanced algorithmic topics.

You have already learned a lot about algorithms,
but there is much more left...

- **Types:** incremental, recursive, D&C, greedy, numerical, exact, approx., randomized, parallel, distributed, ...
- **Analysis:** correctness, runtime, space usage, amortized, expected, optimality, benchmarking, ...
- **Problems:** combinatorial, graphs, geometric





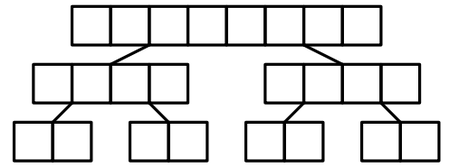


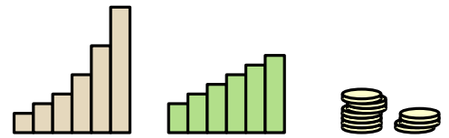
Advanced Algorithms

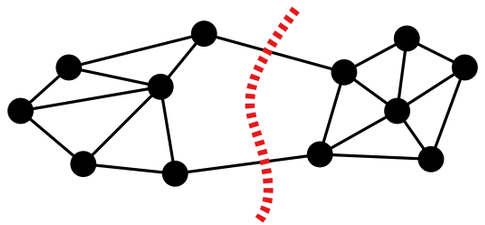
The goal of this course is to offer an overview of advanced algorithmic topics.

You have already learned a lot about algorithms,
but there is much more left...

- **Types:** incremental, recursive, D&C, greedy, numerical, exact, approx., randomized, parallel, distributed, ...
- **Analysis:** correctness, runtime, space usage, amortized, expected, optimality, benchmarking, ...
- **Problems:** combinatorial, graphs, geometric, strings, biological, geographic, ...





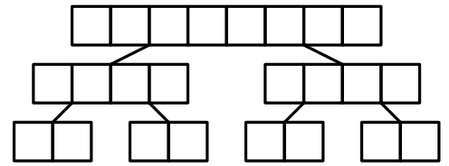


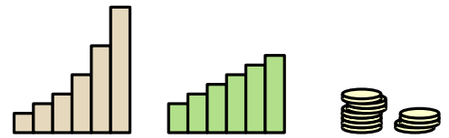
Advanced Algorithms

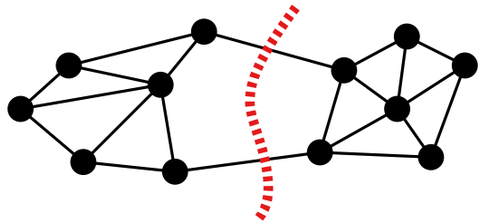
The goal of this course is to offer an overview of advanced algorithmic topics.

You have already learned a lot about algorithms,
but there is much more left...

- **Types:** incremental, recursive, D&C, greedy, numerical, exact, approx., randomized, parallel, distributed, ...
- **Analysis:** correctness, runtime, space usage, amortized, expected, optimality, benchmarking, ...
- **Problems:** combinatorial, graphs, geometric, strings, biological, geographic, ...
- **Data structures:** lists, binary search trees







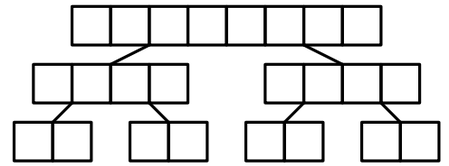
..

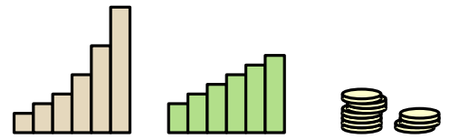
Advanced Algorithms

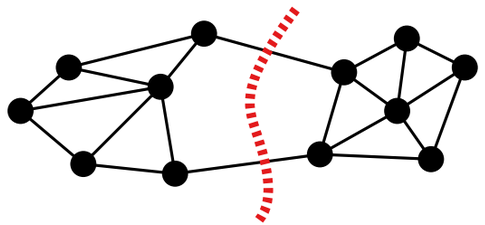
The goal of this course is to offer an overview of advanced algorithmic topics.

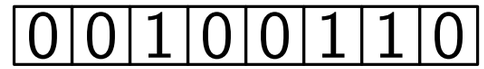
You have already learned a lot about algorithms,
but there is much more left...

- **Types:** incremental, recursive, D&C, greedy, numerical, exact, approx., randomized, parallel, distributed, ...
- **Analysis:** correctness, runtime, space usage, amortized, expected, optimality, benchmarking, ...
- **Problems:** combinatorial, graphs, geometric, strings, biological, geographic, ...
- **Data structures:** lists, binary search trees, dictionaries, succinct, randomized, probabilistic, ...







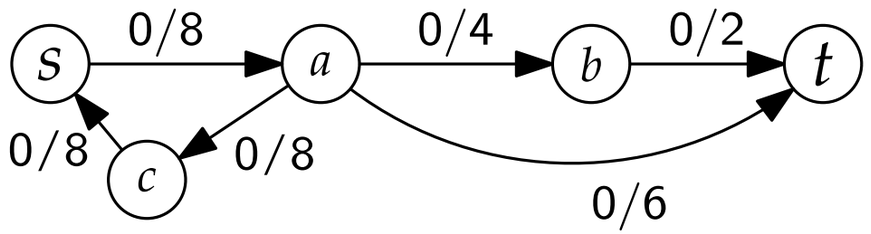


Topics I

- **Better algorithms** for problems you know

Topics I

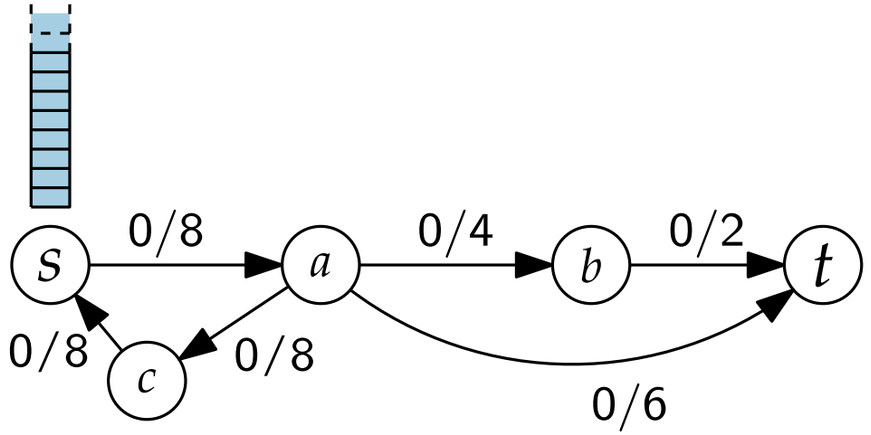
- **Better algorithms** for problems you know
- Maximum flow problem



- Ford–Folkerson algorithm: $\mathcal{O}(|E||f^\star|)$
- Edmonds–Karp algorithm: $\mathcal{O}(|V||E|^2)$
- Push-Relabel algorithm: $\mathcal{O}(|V|^2|E|)$ (or even better)

Topics I

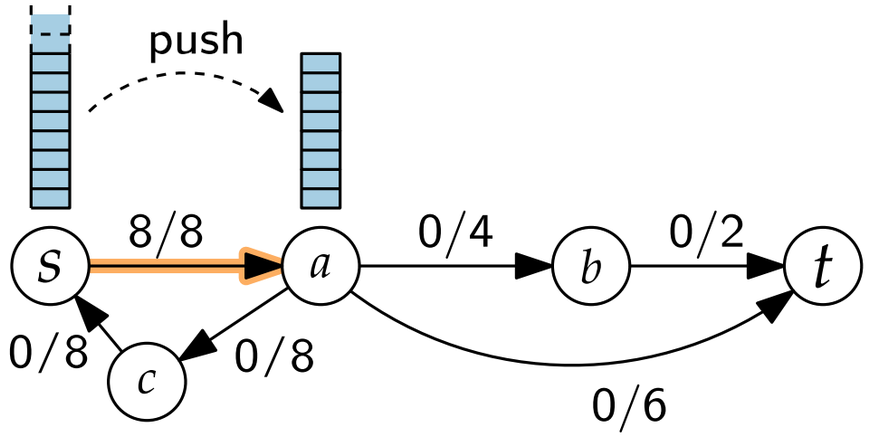
- **Better algorithms** for problems you know
- Maximum flow problem



- Ford–Folkerson algorithm: $\mathcal{O}(|E||f^\star|)$
- Edmonds–Karp algorithm: $\mathcal{O}(|V||E|^2)$
- Push-Relabel algorithm: $\mathcal{O}(|V|^2|E|)$ (or even better)

Topics I

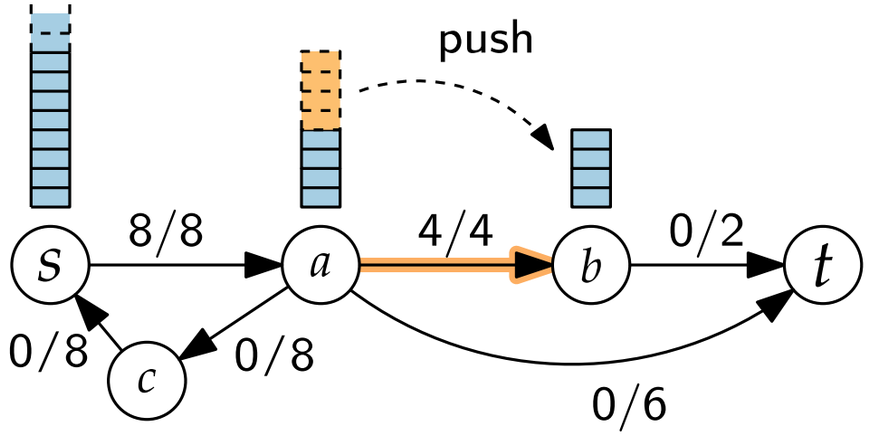
- **Better algorithms** for problems you know
- Maximum flow problem



  - Ford–Folkerson algorithm: $\mathcal{O}(|E||f^\star|)$
  - Edmonds–Karp algorithm: $\mathcal{O}(|V||E|^2)$
  - Push-Relabel algorithm: $\mathcal{O}(|V|^2|E|)$ (or even better)

Topics I

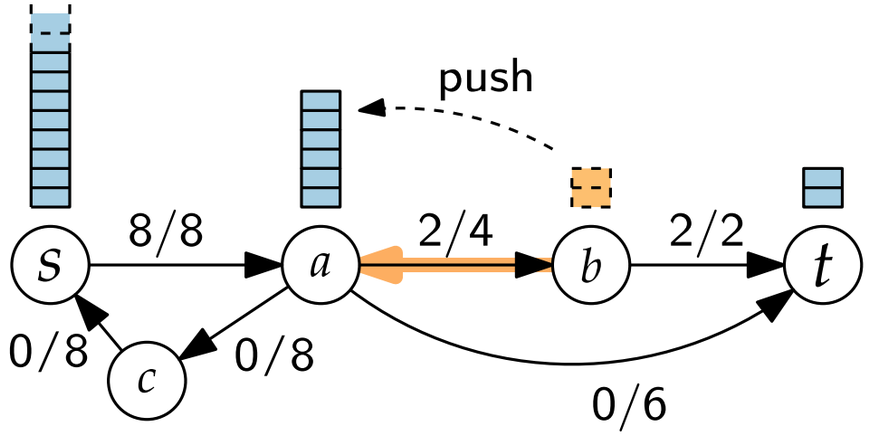
- **Better algorithms** for problems you know
- Maximum flow problem



- Ford–Folkerson algorithm: $\mathcal{O}(|E||f^\star|)$
- Edmonds–Karp algorithm: $\mathcal{O}(|V||E|^2)$
- Push-Relabel algorithm: $\mathcal{O}(|V|^2|E|)$ (or even better)

Topics I

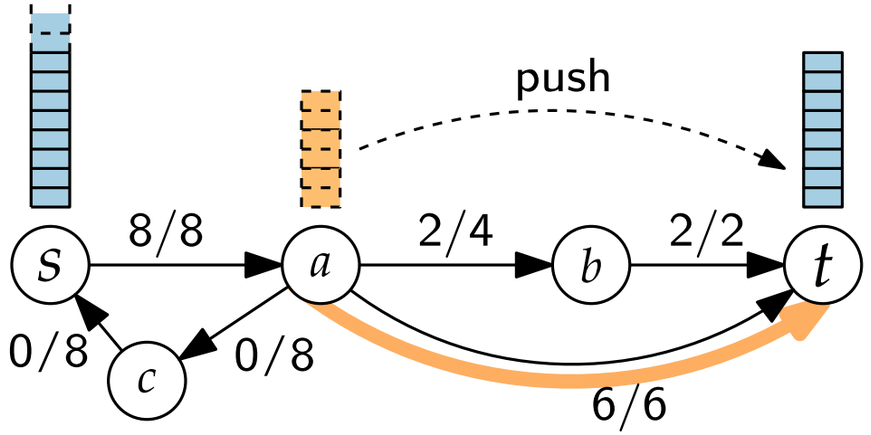
- **Better algorithms** for problems you know
- Maximum flow problem



  - Ford–Folkerson algorithm: $\mathcal{O}(|E||f^\star|)$
  - Edmonds–Karp algorithm: $\mathcal{O}(|V||E|^2)$
  - Push-Relabel algorithm: $\mathcal{O}(|V|^2|E|)$ (or even better)

Topics I

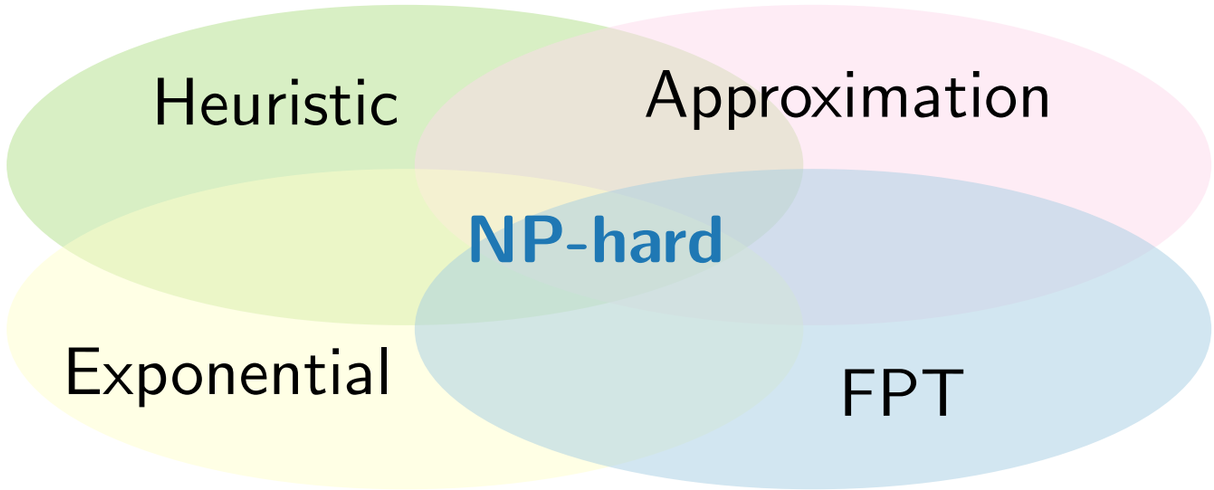
- **Better algorithms** for problems you know
- Maximum flow problem



- Ford–Folkerson algorithm: $\mathcal{O}(|E||f^\star|)$
- Edmonds–Karp algorithm: $\mathcal{O}(|V||E|^2)$
- Push-Relabel algorithm: $\mathcal{O}(|V|^2|E|)$ (or even better)

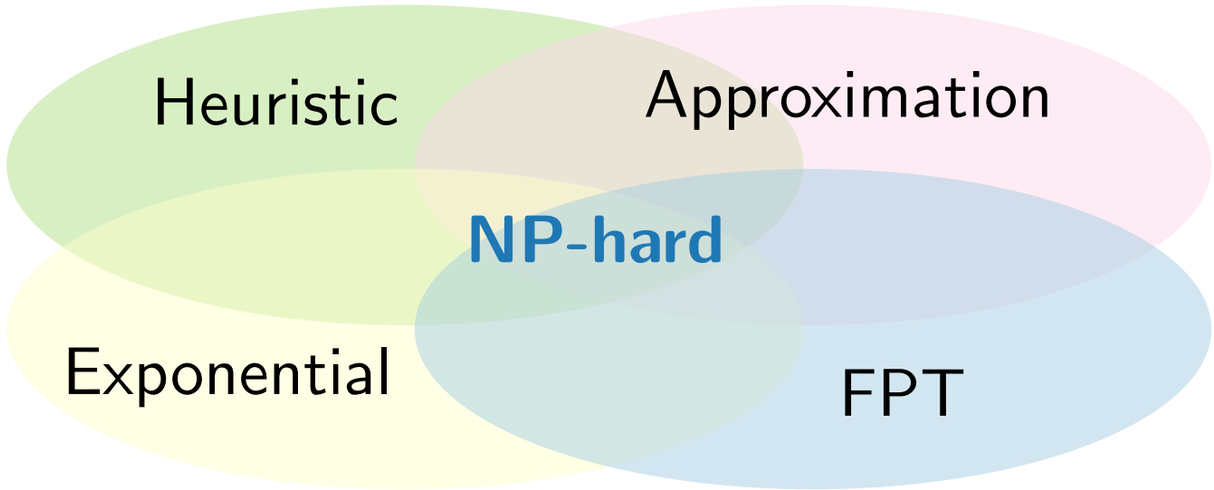
Topics II

- How to deal with **NP-hard problems**



Topics II

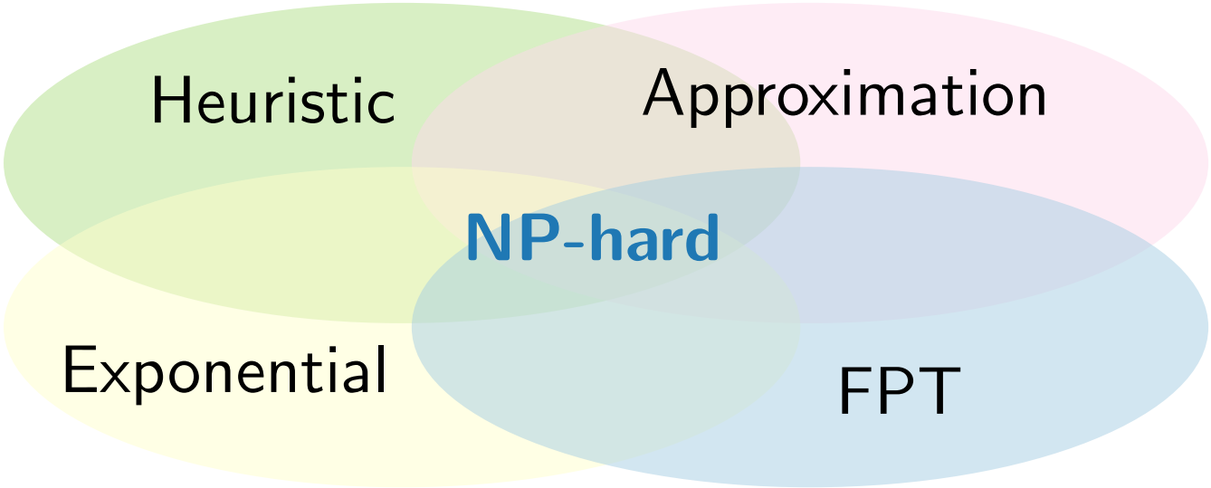
- How to deal with **NP-hard problems**



- Sacrifice quality for speed?
- Can we still compute optimal solutions?

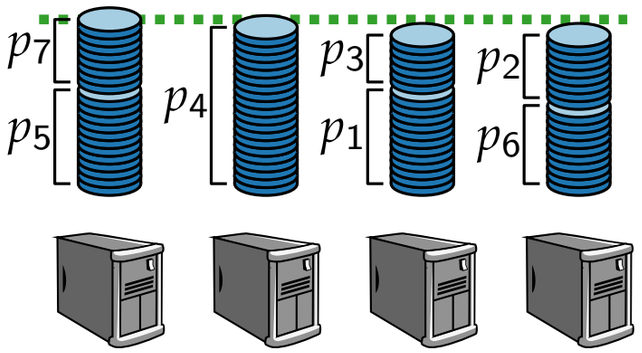
Topics II

- How to deal with **NP-hard problems**



  - Sacrifice quality for speed?
  - Can we still compute optimal solutions?

- Example problem:
  Schedule jobs to machines
  approximating the minimum makespan



Topics III

- **Special** areas

Randomized algorithms

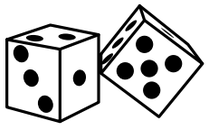
LongestPath	but easy on
is NP-hard	acyclic digraphs

Topics III

- **Special** areas

Randomized algorithms

LongestPath is NP-hard but easy on acyclic digraphs $\Rightarrow$ randomly turn given graph into acyclic digraph



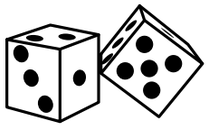
Topics III

- **Special** areas

Randomized algorithms

LongestPath is NP-hard but easy on acyclic digraphs $\Rightarrow$ randomly turn given graph into acyclic digraph

$\Rightarrow$ good idea?



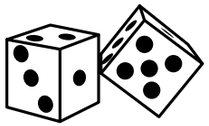
Topics III

- **Special** areas

Randomized algorithms

LongestPath is NP-hard — but easy on acyclic digraphs $\Rightarrow$ randomly turn given graph into acyclic digraph

$\Rightarrow$ good idea?



Also

- Online algorithms
- Computational geometry
- Working with strings

Topics IV

- **Advanced data structures**

Searching for strings

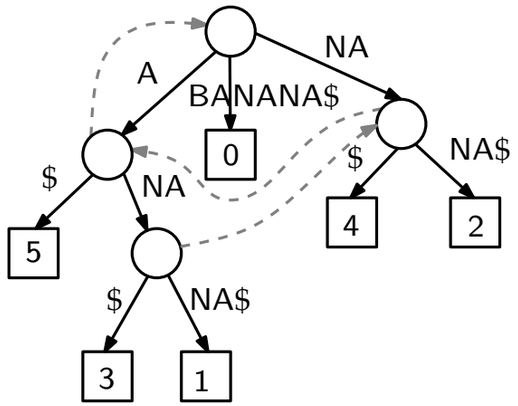
Given text S, how can we efficiently find
all occurrences of pattern P?

Topics IV

- **Advanced data structures**

Searching for strings

Given text S, how can we efficiently find all occurrences of pattern P?



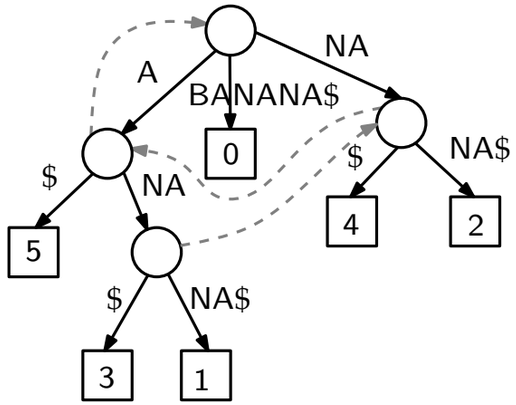
- Suffix trees
- Invest in preprocessing to be faster than full parse

Topics IV

- **Advanced data structures**

Searching for strings

Given text S, how can we efficiently find all occurrences of pattern P?



- Suffix trees
- Invest in preprocessing to be faster than full parse

Also

- Succinct data structures
- Splay trees

Lectures

- Johannes Zink

- Email: johannes.zink@uni-wuerzburg.de
- Office: Room 01.007, Building M4 (next to computer science building)

Lectures

- Johannes Zink

- Email: johannes.zink@uni-wuerzburg.de
- Office: Room 01.007, Building M4 (next to computer science building)

- In-person lectures Wed, 14:15–15:45, ÜR I
- With time for questions and discussions
- 12 or 13 lectures
- Old videos from 2020 will be made available on WueCampus

Tutorials

- Oksana Firman

- Email: oksana.firman@uni-wuerzburg.de
- Office: Room 01.005, Building M4

- In-person tutorials Mon, 16:00–17:30, HS 4, Physics building
- With time for questions and discussions
- 11 or 12 exercise sheets

Tutorials

- Oksana Firman

- Email: oksana.firman@uni-wuerzburg.de
- Office: Room 01.005, Building M4

- In-person tutorials Mon, 16:00–17:30, HS 4, Physics building
- With time for questions and discussions
- 11 or 12 exercise sheets

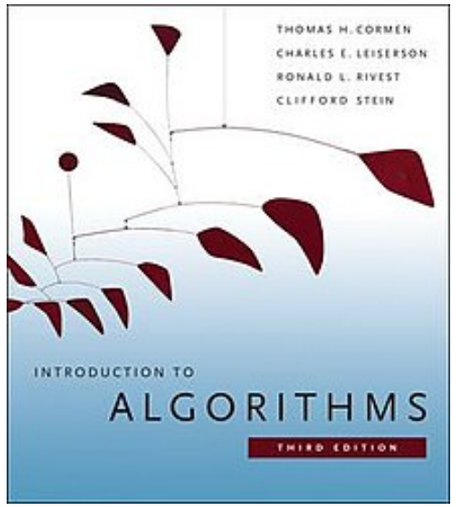
Exercise sheets.

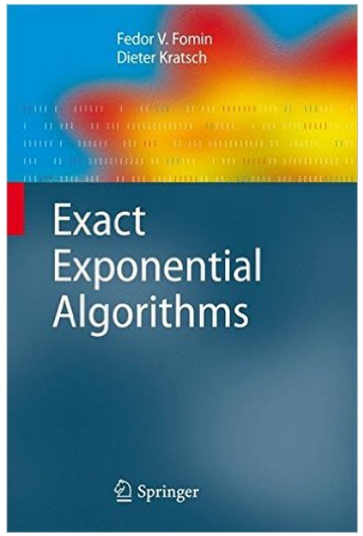
- Weekly exercise sheets, $\approx$ 20 points/sheet
- Scoring 50 % of the points grants a bonus of 0.3 to the final grade of the exam (if one passes)
- Released at the lecture day (Wed)
- Submission deadline next lecture (Wed, 14:15)
- Digital submission as pdf; recommended to use our LaTeX template
- Submission in teams of two . . .
- . . . in English (preferred) or German

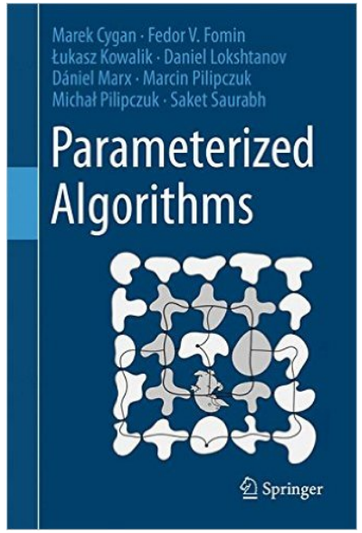
Exam

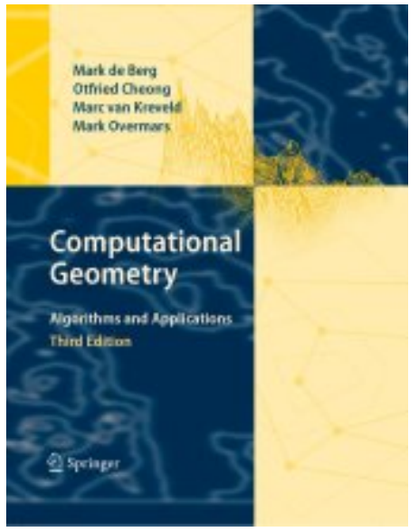
- Oral exam
- $\approx$ 20min
- Bonus for points on the exercise sheets (see previous page)
- Date will be announced during the semester
- Don't forget to register in WueStudy:
  "Ausgewählte Kapitel der ..."

Literature



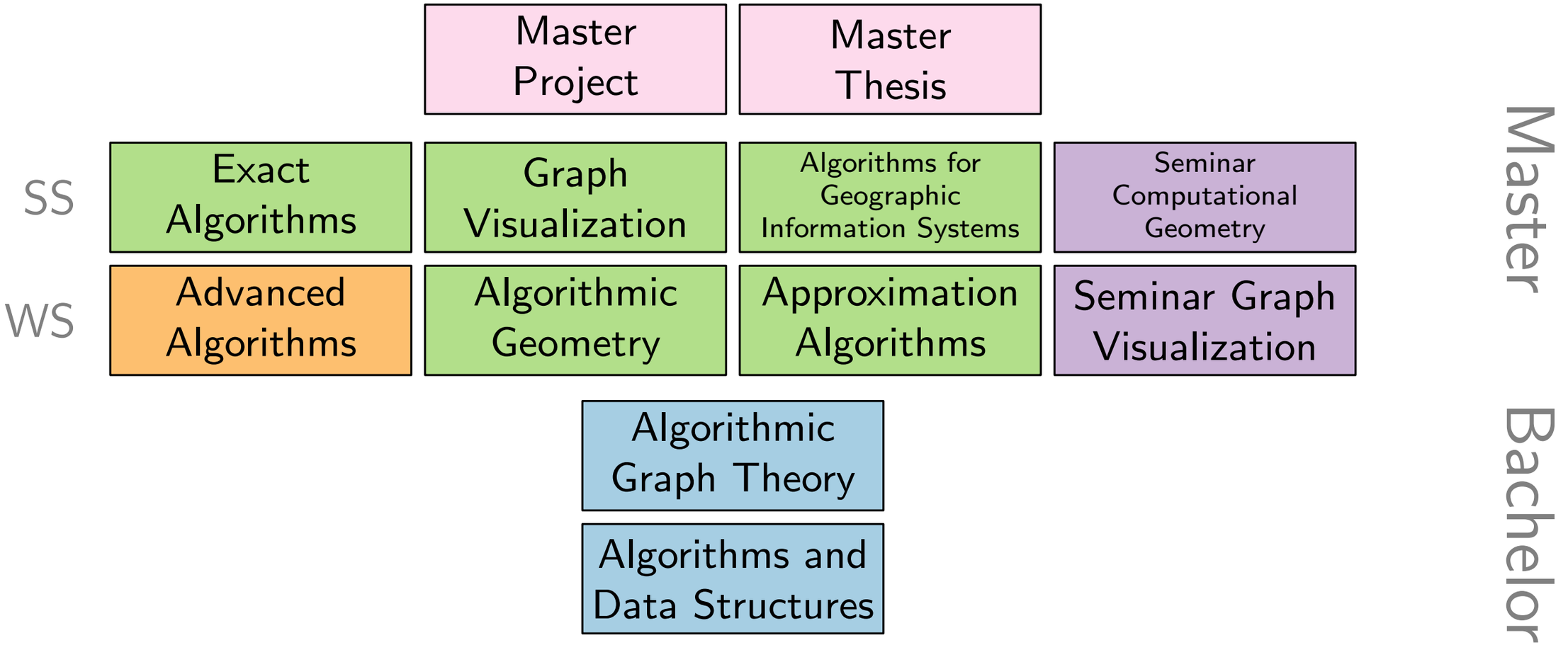






- Sources at the end of every lecture
- Links to further interesting stuff

Our Lectures and Seminars



Thanks

Material and slides provided in this lecture have been compiled
by many different people. Special thanks to:

Jonathan Klawitter, Boris Klemz, Steven Chaplick,
Thomas van Dijk, Philipp Kindermann, Joachim Spoerhase,
Sabine Storandt, Dorothea Wagner, Tim Hegemann,
Alexander Wolff, ...