

# Algorithms Lab

Prof. Dr. Angelika Steger

Prof. Dr. Emo Welzl

Prof. Dr. Peter Widmayer

# Objective: Master of Algorithms

**Design efficient algorithms to solve “real world” problems.**

(real world = toy world)

Problem given as a text/story. Your task includes:

- appropriate problem modeling,
- choice of suitable algorithms, and
- implementation.

**Note:** usually the hard part is to come up with the algorithm, not implementing it. Most problems can be solved with less than 100 lines of (well formatted) C++ code.

# Objective: Master of Algorithms

**Design efficient algorithms to solve “real world” problems.**  
(real world = toy world)

Problem given as a text/story. Your task includes:

- appropriate problem modeling,
- choice of suitable algorithms, and
- implementation.

**Note:** usually the hard part is to come up with the algorithm, not implementing it. Most problems can be solved with less than 100 lines of (well formatted) C++ code.

# Objective: Master of Algorithms

**Design efficient algorithms to solve “real world” problems.**  
(real world = toy world)

Problem given as a text/story. Your task includes:

- appropriate problem modeling,
- choice of suitable algorithms, and
- implementation.

**Note:** usually the hard part is to come up with the algorithm, not implementing it. Most problems can be solved with less than 100 lines of (well formatted) C++ code.

# Objective: Master of Algorithms

**Design efficient algorithms to solve “real world” problems.**  
(real world = toy world)

Problem given as a text/story. Your task includes:

- appropriate problem modeling,
- choice of suitable algorithms, and
- implementation.

**Note:** usually the hard part is to come up with the algorithm, not implementing it. Most problems can be solved with less than 100 lines of (well formatted) C++ code.

# Course format

- This is a lab → most time spent working individually
- **Tutorial:** Wednesday 17 – 19
  - background and technical issues related to programming environment and software libraries
  - solutions to problems from preceding week
  - recap of known algorithmic concepts with examples (also teach a few new ones, but focus on applications rather than theory)
- **Problem of the week:** Monday 17 – 19 (more on this later)  
Exam like conditions: one problem, two hours to solve
- **Office hours:** Wednesday 19– (after the tutorial)

# Course format

- This is a lab → most time spent working individually
- **Tutorial:** Wednesday 17 – 19
  - background and technical issues related to programming environment and software libraries
  - solutions to problems from preceding week
  - recap of known algorithmic concepts with examples (also teach a few new ones, but focus on applications rather than theory)
- **Problem of the week:** Monday 17 – 19 (more on this later)  
Exam like conditions: one problem, two hours to solve
- **Office hours:** Wednesday 19– (after the tutorial)

# Course format

- This is a lab → most time spent working individually
- **Tutorial:** Wednesday 17 – 19
  - background and technical issues related to programming environment and software libraries
  - solutions to problems from preceding week
  - recap of known algorithmic concepts with examples (also teach a few new ones, but focus on applications rather than theory)
- **Problem of the week:** Monday 17 – 19 (more on this later)  
Exam like conditions: one problem, two hours to solve
- **Office hours:** Wednesday 19– (after the tutorial)



# Course format

- This is a lab → most time spent working individually
- **Tutorial:** Wednesday 17 – 19
  - background and technical issues related to programming environment and software libraries
  - solutions to problems from preceding week
  - recap of known algorithmic concepts with examples (also teach a few new ones, but focus on applications rather than theory)
- **Problem of the week:** Monday 17 – 19 (more on this later)  
Exam like conditions: one problem, two hours to solve
- **Office hours:** Wednesday 19– (after the tutorial)

# Prerequisites

See the course website:

<http://www.cadmo.ethz.ch/education/lectures/HS15/algolab/prereqs>

Most important:

**Strategies** Brute force, greedy, divide & conquer, dynamic programming, backtracking

**Data structures** Array, stack, FIFO queue, tree, heap, priority queue

**Graph algorithms** DFS, BFS, minimum spanning tree, Eulerian tour

**Graph concepts** Directed graph, coloring, matching, topological sorting

Occasionally we do a recap, but we do *not* give full explanations.

## Three parts:

**Fundamentals** BFS, DFS, greedy, divide & conquer, elementary geometric computing in CGAL, elementary graph representations and algorithms in BGL

**Advanced Algorithms** Dynamic programming, network flows and applications, LP/QP, Delaunay triangulations

**Exam preparation** select & combine from the above

## Three types of problems / three libraries:

ACM C++ standard library

BGL Boost Graph Library

CGAL Computational Geometry Algorithms Library

# Synopsis

## Three parts:

**Fundamentals** BFS, DFS, greedy, divide & conquer, elementary geometric computing in CGAL, elementary graph representations and algorithms in BGL

**Advanced Algorithms** Dynamic programming, network flows and applications, LP/QP, Delaunay triangulations

**Exam preparation** select & combine from the above

## Three types of problems / three libraries:

ACM C++ standard library

BGL Boost Graph Library

CGAL Computational Geometry Algorithms Library

## Three parts:

**Fundamentals** BFS, DFS, greedy, divide & conquer, elementary geometric computing in CGAL, elementary graph representations and algorithms in BGL

**Advanced Algorithms** Dynamic programming, network flows and applications, LP/QP, Delaunay triangulations

**Exam preparation** select & combine from the above

## Three types of problems / three libraries:

ACM C++ standard library

BGL Boost Graph Library

CGAL Computational Geometry Algorithms Library

## Three parts:

**Fundamentals** BFS, DFS, greedy, divide & conquer, elementary geometric computing in CGAL, elementary graph representations and algorithms in BGL

**Advanced Algorithms** Dynamic programming, network flows and applications, LP/QP, Delaunay triangulations

**Exam preparation** select & combine from the above

## Three types of problems / three libraries:

**ACM** C++ standard library

**BGL** Boost Graph Library

**CGAL** Computational Geometry Algorithms Library

# Problems

- Every Wednesday: new set of problems
- Automated grading/feedback by an online judge
- Discussion: next Wednesday
- We present solutions for selected problems.

## Testat

No testat. – However ...

6 ECTS credits correspond to 180 working hours.

# Problem of the week

- Every Monday at 17:00, we post a special exam-style problem.
- You have to solve this problem within the next two hours.
- To motivate you, we will keep a scoreboard of what you achieved during these 2 hours.
- Use this opportunity to test your fitness for the exam.
- To really assess your skills, only use resources also available during the exam.

NEW: hidden test sets — only partial feedback on the judge (typically about 80/100 points)



# Problem of the week

- Every Monday at 17:00, we post a special exam-style problem.
- You have to solve this problem within the next two hours.
- To motivate you, we will keep a scoreboard of what you achieved during these 2 hours.
- Use this opportunity to test your fitness for the exam.
- To really assess your skills, only use resources also available during the exam.

NEW: **hidden test sets** — only partial feedback on the judge (typically about 80/100 points)

# Problem of the week

- Every Monday at 17:00, we post a special exam-style problem.
- You have to solve this problem within the next two hours.
- To motivate you, we will keep a scoreboard of what you achieved during these 2 hours.
- Use this opportunity to test your fitness for the exam.
- To really assess your skills, only use resources also available during the exam.

NEW: hidden test sets — only partial feedback on the judge  
(typically about 80/100 points)

# Problem of the week

- Every Monday at 17:00, we post a special exam-style problem.
- You have to solve this problem within the next two hours.
- To motivate you, we will keep a scoreboard of what you achieved during these 2 hours.
- Use this opportunity to test your fitness for the exam.
- To really assess your skills, only use resources also available during the exam.

**NEW: hidden test sets** — only partial feedback on the judge (typically about 80/100 points)

# Computer Rooms

Solutions to problems (incl. the problem of the week) can be handed in from anywhere (you just need an Internet connection).

We have reserved computer rooms on Mondays which you can use when solving the problem of the week.

- In CAB: CAB H 56, CAB H 57 (assistants drop by here)
- In HG: HG E 26.1
- Everywhere else on your own laptop (+Internet connection)

## Grade

Grade based only on exam.

- 2 sessions of 6 hours each
- HG computer rooms
- Submission/judging of programs exactly as in semester
- **NEW: hidden test sets**
- Very similar to the potw
- Open-book, no electronics, no Internet
- No repetition!

# How to get help

If you cannot solve a problem, you have two options.

- Your best bet for quick help are our **forums**, where other students can help you out.
- Alternatively, use the office hours after the tutorial.
- **Important:** try to solve the problems on your own. In the exam you will not have access to the forums.

ACM Rajko Nenadov, Nemanja Škorić

BGL Andreas Baertschi, Daniel Graf

CGAL Michael Hoffmann, Antonis Thomas

## Contact

[algolab@lists.inf.ethz.ch](mailto:algolab@lists.inf.ethz.ch)

- Only use the mailing list for technical problems or if you think there are issues with a problem statement/specification!
- All other issues should be discussed on the forums.

ACM Rajko Nenadov, Nemanja Škorić

BGL Andreas Baertschi, Daniel Graf

CGAL Michael Hoffmann, Antonis Thomas

## Contact

[algolab@lists.inf.ethz.ch](mailto:algolab@lists.inf.ethz.ch)

- Only use the mailing list for technical problems or if you think there are issues with a problem statement/specification!
- All other issues should be discussed on the forums.



## There you can:

- download slides of the tutorials,
- download problem sheets,
- submit solutions, and
- discuss problems with your colleagues in the forums.

## Login

- <https://moodle-app2.let.ethz.ch/login>
- NETHZ account / NETHZ password
- Enrolment Key (for the judge) *predictivealgorithm*

# Break

After the break:

- Sample problem
  - Show you the complete procedure of solving & submitting a problem to the judge
- Course website
  - Forum etiquette