

If you want to participate, turn on your camera.

# Organization

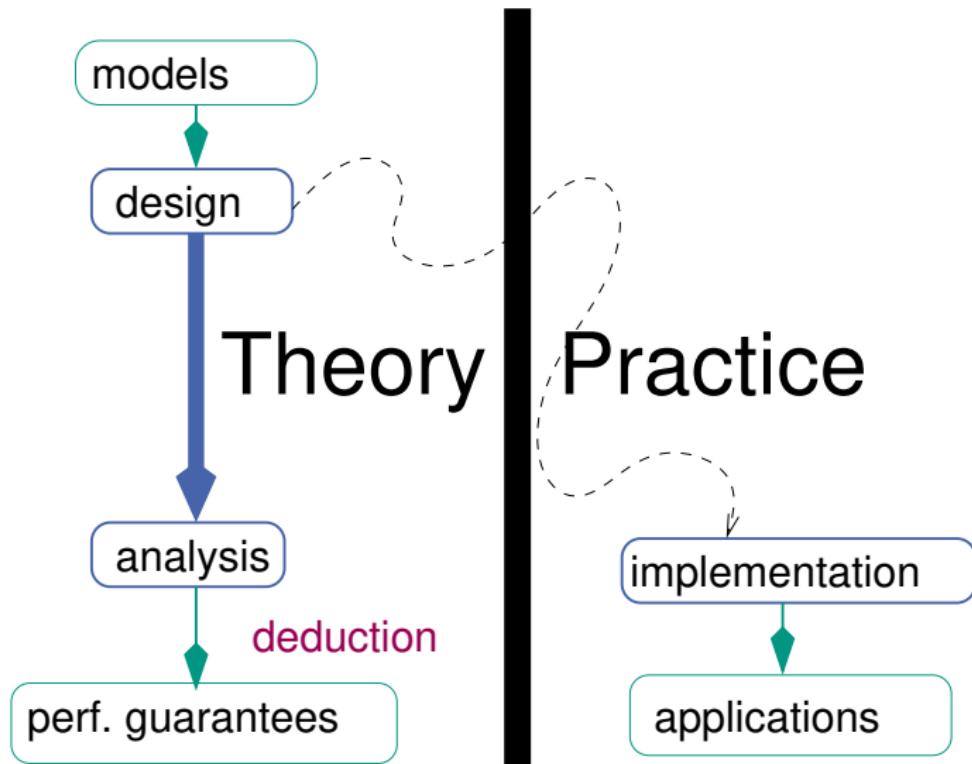
## Lecturer



### Christian Schulz

- University of Heidelberg
- Mail:  
[christian.schulz@informatik.uni-heidelberg.de](mailto:christian.schulz@informatik.uni-heidelberg.de)
- Room 1/328

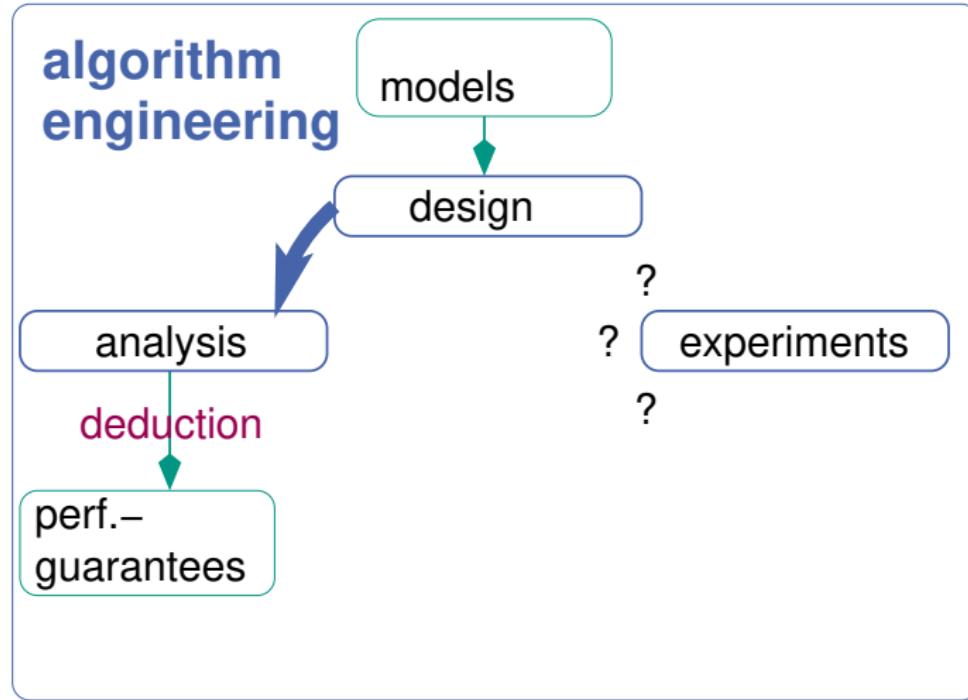
# (Caricatured) Traditional Algorithm Theory



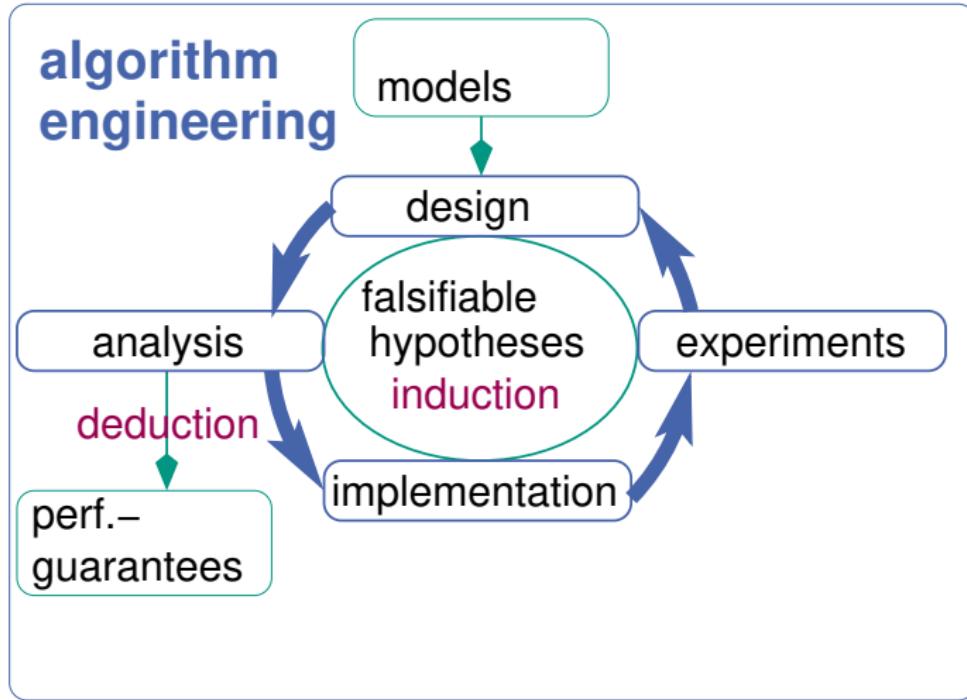
# Gaps Between Theory & Practice

Theory	↔	Practice		
simple		<b>appl. model</b>		complex
simple		<b>machine model</b>		real
complex		<b>algorithms</b>		simple
advanced		<b>data structures</b>		arrays,...
worst case		<b>complexity measure</b>		inputs
asympt.		<b>efficiency</b>		42% constant factors

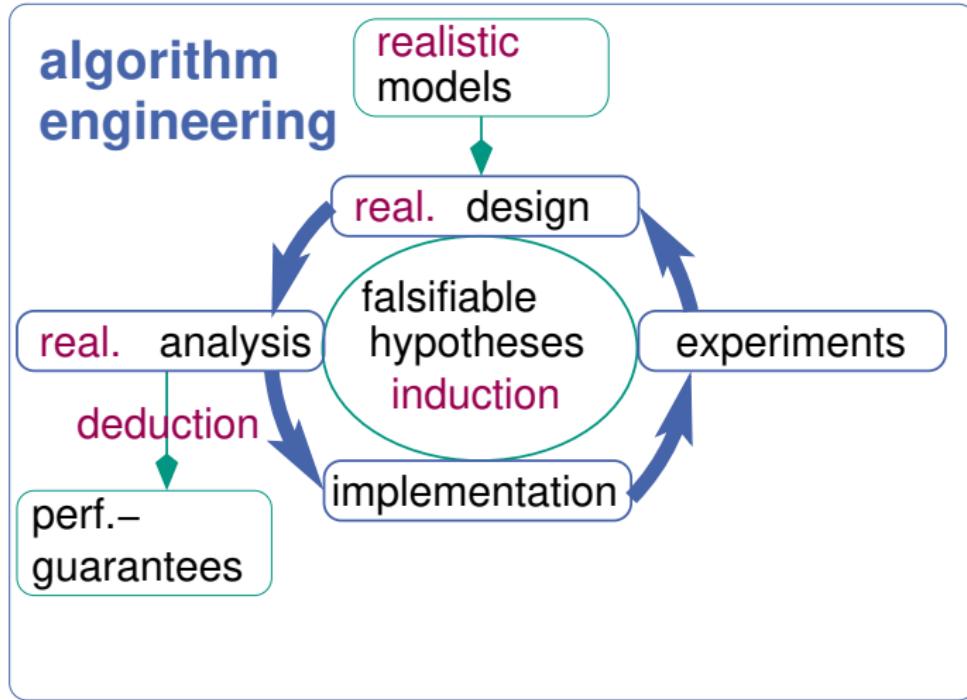
# Algorithmics as Algorithm Engineering



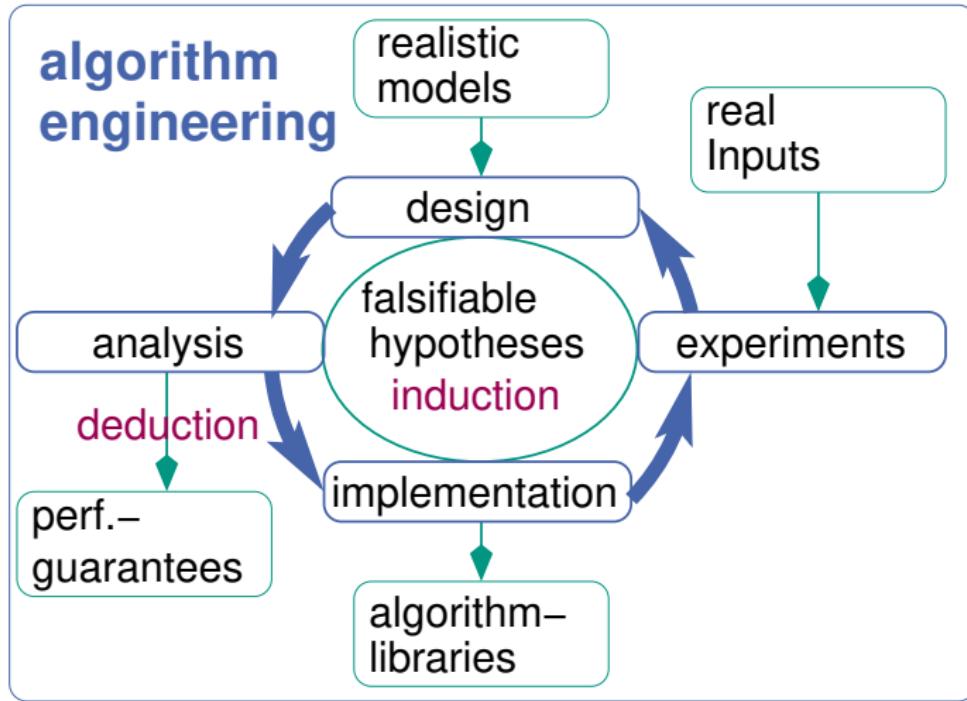
# Algorithmics as Algorithm Engineering



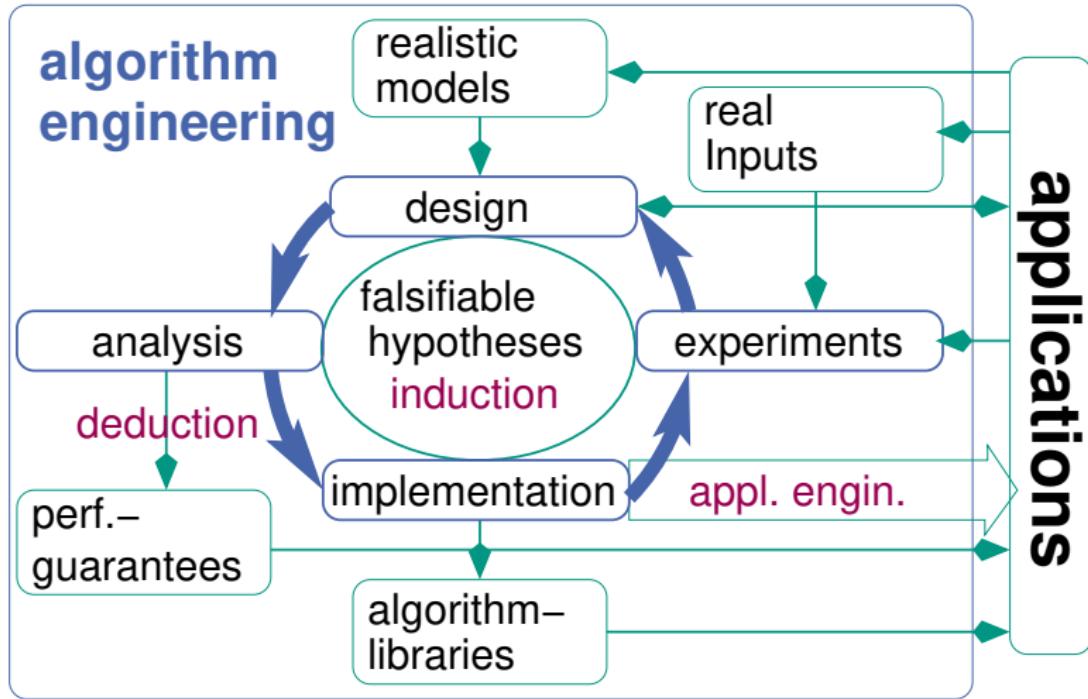
# Algorithmics as Algorithm Engineering



# Algorithmics as Algorithm Engineering

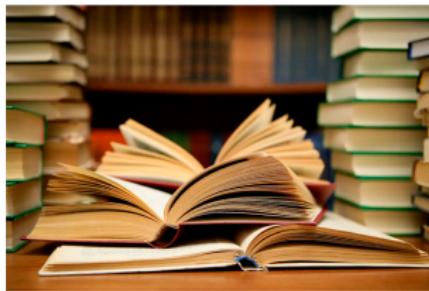


# Algorithmics as Algorithm Engineering



# Organization

- 8 LP, 6 SWS (adv), 2LP+4LPFÜK, 4 SWS (beginner)
- 240h workload (adv), 180h workload (beginner)
- The project takes place **online**
- We will have **weekly** meetings
- We can only take 2-3 students.
- The project will conclude with a 15min presentation
- You already need programming very good skills in C++ or C. Algorithms will be programmed in that language. If you feel uncomfortable in C++ don't take the course.



# Organization

The course teaches you how to do algorithm engineering RESEARCH.

- The programming projects are **research oriented**, i.e. you will work on a topic that may turn out to become a publication.
- We will (often) work with (international) colleagues
- You have to hand in a written report in the end.
- The programming project is meant as a preparation for a thesis project! If you finish the project successfully then, you can and *should* continue to work on the topic for your **final thesis project**



# Organization

Send me your transcripts  
and  
a list of topics that you like!



# Topics

# Differential Privacy

~~> reproducing differential privacy result of  $k$ -means problems  
in particular implement DP Lloyds algorithm

<https://dl.acm.org/doi/pdf/10.1145/3133201>

# Machine Learning for Greedy Algorithms

~~ whiteboard

# Python Interfaces and Usability for Open Source Projects

~~> beginner only

# Connected Partitions / Fixed Vertices in KaHIP

~~> whiteboard

# Budgeted Dynamic Betweenness Maximization

~~ whiteboard

# Dynamic MST

~~ whiteboard

# MinMax Graph Partitioning

~~ whiteboard

# **State-of-the-Art Max Clique Solver**

~~ whiteboard

# Transport Logistics with PTV

~~ whiteboard