# inf451 - Complexity Theory

<table>
<thead>
<tr>
<th>Module name</th>
<th>Complexity Theory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module code</td>
<td>inf451</td>
</tr>
<tr>
<td>ECTS credit points</td>
<td>6.0 KP</td>
</tr>
<tr>
<td>Workload</td>
<td>180 h</td>
</tr>
<tr>
<td>Used in degree programmes</td>
<td>Master's Programme Computing Science &gt; Mastermodule</td>
</tr>
<tr>
<td>Contact person</td>
<td></td>
</tr>
<tr>
<td>module responsibility</td>
<td></td>
</tr>
<tr>
<td>authorized examiners</td>
<td></td>
</tr>
<tr>
<td>Eike Best</td>
<td></td>
</tr>
<tr>
<td>Die im Modul Lehrenden</td>
<td></td>
</tr>
</tbody>
</table>

## Prerequisites

Skills to be acquired in this module

This module covers the computational complexity of algorithms. Complexity considerations are concerned with the time, the memory, and the parallelism required or allowed, for solving an algorithmic problem. In particular, one is interested in lower and/or upper time and space bounds, and in approximative investigations providing information about entire classes of algorithms. For any concrete problem, complexity theory aims at being able to find out which class it belongs to, and thus estimating the cost of the most efficient methods of solving it.

Methods taught in this module are general, not depending on any particular algorithmic model or chosen programming language.

### Professional competence

The students:

- use Turing machines and variants thereof
- define time, memory, and processor requirements of algorithmic problems
- specify the most relevant complexity classes
- estimate the computational complexity of the most important problems

### Methodological competence

The students:

- analyse the complexity of algorithms
- apply techniques of simulation, reduction, and diagonalisation
- compare new problems in terms of complexity

### Social competence

The students:

- present proof sketches, proofs, and algorithmic solutions in front of an audience

## Module contents

- Mathematical foundations
- Turing machines and register machines
- Space and time hierarchies, equivalence and hierarchy theorems
- Complexity classes: P, NP, NPC, PSPACE, and others
- Alternating automata and polynomial time hierarchy
- Circuit complexity

Recommended reading

- Eike Best: Skript zur Vorlesung (2015)

Links

Language of instruction: German

Duration (semesters): 1 semester

Module frequency: unregelmäßig

Module capacity: Unlimited

Module level: AC (Aufbaucurriculum / Composition)

Modulart: je nach Studiengang Pflicht oder Wahlpflicht

Lern-/Lehrform / Type of program

Vorkenntnisse / Previous knowledge

Examination
- Examination periods: At the end of the lecture period
- Type of examination: exercises and oral exam

Course type | Comment | SWS | Course frequency | Workload attendance
---|---|---|---|---
Lecture | | 2 | | 28 h
Exercises | | 2 | | 28 h
Total attendance time for module: 56 h