Introduction, administrivia

Data Structures and Algorithms for Computational Linguistics III
ISCL-BA-07

Çağrı Çöltekin
/tʃaːrˈɯ tʃœltecˈɪn/
ccoltekin@sfs.uni-tuebingen.de

University of Tübingen
Seminar für Sprachwissenschaft

Winter Semester 2024–2025

version: c8a5ad5 @2023-10-23

What is this course about?

- An intermediate-level course on programming
- Algorithms: (good) solutions to programming problems
- Data structures: (efficient) ways to organize/store information

Prerequisites:

- Data Structures and Algorithms for CL I
- Data Structures and Algorithms for CL II

Module: ISCL-BA-07, Advanced Programming

What is in this course?

A bird's eye view

Introductory lectures on

- Some fundamental data structures: arrays, queues, stacks, trees, ...
- Some fundamental algorithms: searching, sorting, pattern matching, graph algorithms
- Analysis of algorithms
- Finite state automata
- Parsing

Why study algorithms?

- It is one of the fundamental topics in computer science: an algorithm is the way you instruct a computer to do things
- Knowing a clever, efficient solution to one problem helps designing good solutions for other, related problems
- Learning basic algorithmic techniques makes you a better programmer
- Designing good algorithms is an intellectual challenge
- The most popular interview questions for programming jobs are about algorithms

Course overview

- Lectures (Lothar-Meyer-Bau 301)
  - Monday 14:15-15:45
  - Wednesday 14:15-15:45
- Lab: Lothar-Meyer-Bau 301 (?), Friday 14:15-17:45
- Tutors:
  - Darja Jepifanova
  - Giulio Cusenza
- Public course website: `https://dsacl3-2024.github.io/`
- Moodle: `https://moodle.zdv.uni-tuebingen.de/course/view.php?id=58`
- GitHub: `https://github.com/dsacl3-2024/dsacl3`

Literature

- **goodrich2013**. **goodrich2013** (**goodrich2013**)
  - Available through university library (online version): `https://ebookcentral.proquest.com/lib/unitueb/detail.action?docID=4946360`
  - Website of the book contains source code, hints, examples: `http://bcs.wiley.com/he-bcs/Books?action=index&bcsId=8029&itemId=1118290275`
- **jurafsky2009**. **jurafsky2009** (**jurafsky2009**)
  - Draft chapters of 3rd edition is available at `https://web.stanford.edu/~jurafsky/slp3/`
- Course notes will be provided for some topics

Coursework and evaluation

- Reading material for most lectures
- Weekly assignments: ungraded, but **required**: For successful completion of the practical part of the class, you have to complete all the assignments, and at least 80% of them has to be on time.
- Final (written) exam (70 %)
- Final project (30 %)
- Attendance is not required, but you are unlikely to pass without regular attendance

Assignments

- Assignments in Python
- Only online submissions through GitHub
- The assignments can be done in pairs (strongly recommended – knowing your classmates, and learning from them, is an important part of the university experience/education)
- This means **working together on the whole exercise**, not sharing parts of an assignment and working on them independently
- You can pair with the same person only once
- We will have a match-making mechanism
- See course page for more information

Topics at a glance

- A recap of what you should already know: arrays, lists, maps, queues, stacks, iteration, recursion, binary search, ...
- Algorithmic analysis
- Common algorithmic patterns: brute force, greedy, divide and conquer, dynamic programming, ...
- Sorting
- Trees
- Priority queues, heaps
- Hashing
- Graphs, graph algorithms
- Pattern matching
- Tries
- Finite state automata and regular expressions
- Finite state transducers
- Parsing

Final remarks

- Please do not be shy, ask your questions during the lectures
- Please take the assignments seriously, learning programming requires practice
- Next:
  - a recap of basic data structures and algorithms
  - assignment 0 (also a Python tutorial ?)

- Time for your questions

Acknowledgments, credits, references

- Some of the slides are based on the previous year's course by Corina Dima.