Introduction, administrivia

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

Çağrı Çöltekin
/tʃaːɾˈɯ tʃœltecˈın/
ccoltekin@sfs.uni-tuebingen.de

University of Tübingen
Seminar für Sprachwissenschaft

Winter Semester 2021–2022

version: f63895e @2021-10-25

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 (all online):
  - Monday 14:15-15:45 (lecture)
  - Wednesday 12:15-14:45 (lecture)
  - Friday 14:15-17:45 (lab)
- Tutors:
  - Siena Biales
  - Klara Lennermann
- Public course website: `https://dsacl3-2021.github.io/`
- Moodle: `https://moodle.zdv.uni-tuebingen.de/course/view.php?id=2057`
- GitHub: `https://github.com/dsacl3-2021/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 editon 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
- 6 programming assignments (approximately every two weeks)
  - The best 5 assignments count (as 60 % of your total grade)
  - NEW Each assignment is paired with a short quiz
- Final (written) exam (40 %)
- Attendance is not required, but you are unlikely to pass without regular attendance

Assignments

- Assignments in Python
- Only online submissions through GitHub
- Up to one week late = half the points, more than one week late = 0 points
- Solutions will be discussed after the late-assignment deadline
- 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 all exercises**, not sharing and parts of an assignment and working on them independently
- 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
- Regular expressions and finite state automata
- 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
  - 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.