

Lecture #0 Administrative Elements

Fundamental Algorithms

Rodica Potolea, Camelia Lemnaru and Ciprian Oprea

Technical University of Cluj-Napoca
Computer Science

October 2024

Core Team

- English track + Romanian A track
 - Rodica Potolea
 - Professor, Computer Science Department
 - Room C09
 - Rodica.Potolea@cs.utcluj.ro
- Romanian B track
 - Camelia Lemnaru
 - Room M03
 - Camelia.Lemnaru@cs.utcluj.ro
 - Ciprian Oprisa
 - Ciprian.Oprisa@cs.utcluj.ro
- Core TAs: Raluca Portase, Richard Ardelean, Vlad Negru

Structure of the course

- Lectures
 - <https://moodle.cs.utcluj.ro/course/view.php?id=693>
 - Slides + discussions + pseudocode
 - Open course with Q&A sessions: ask questions when in doubt!
- Tutorials (Seminars)
 - Problem solving – analysis and design, evaluation, comparisons
 - Pseudocode
- Laboratory sessions
 - Problem solving (alg. implementation, testing and evaluation)
 - C/C++
 - same content for all, each group with a different faculty member or (former) PhD/graduate/master student



Lab sessions info – Romanian Tracks

Group	Enrollment Key	TA
30221	Group@30221	Paul Helmer
30222	Group@30222	Anamaria Migea
30223	Group@30223	Alex Lapusan
30224	Group@30224	Richard Ardelean
30225	Group@30225	Vicentiu Cojocaru
30226	Group@30226	Raluca Portase
30227	Group@30227	Raluca Portase
30228	Group@30228	Tudor Trasculescu
30229	Group@30229	Remus Petrache
302210	Group@302210	Vasile Suciu
32721 (CSC)	Group@32721	Anca Elena Iordan

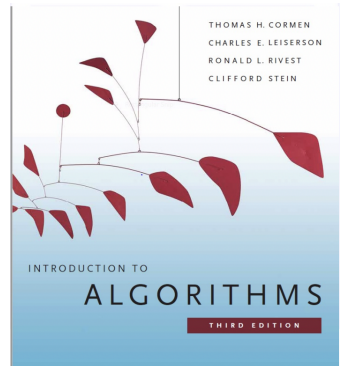


Lab sessions info – English Track

Group	Enrollment Key	TA
30421	Group@30421	Mihai Feier
30422	Group@30422	Vlad Andrei Negru
30423_1	Group@30423_1	Cristian Lungu
30423_2	Group@30423_2	Vlad Andrei Negru
30424	Group@30424	Vlad Andrei Negru
30425	Group@30425	Richard Ardelean

"The Bible"

- Cormen, Leiserson, Rivest, (Stern), "Introduction to Algorithms", first edition 1990 (second/third edition 2001), fourth edition 2021, MIT Press
- e-copy - moodle
- hard copy - CS Department Library, Baritui 26-28, Room M04





Evaluation (1)

- Course Quizzes
 - Multiple quizzes during lectures
 - Target info in the current course and ALL the things discussed up to that point
 - Delivered on Moodle/in class
 - 5% of the Final Grade; CANNOT retake the quizzes
- Mid-Term Exam
 - During lecture / separate sessions – TBD
 - 15% of the Final Grade



Evaluation (2)

- Hands on evaluation / laboratory assignments
 - Stay in your group, presence **mandatory**
 - 10 assignments
 - Every (other) lab deadline on an assignment (various thresholds; we encourage evolution & knowledge/skills accumulation)
 - Late assignments policy:
 - Some assignments can be submitted 1 week late: 80% of max grade
 - More than 1 week, no grade (0) on the given assignment
 - Plagiarism policy – **0 tolerance!!! Don't even try!**
 - 30% in the Final Grade (need grade of 5 or more to be allowed to take the Final Exam)

Evaluation (3)

- Course Quizzes + MT
 - 20% of the Final Grade
- Hands on evaluation / laboratory assignments
 - 30% in the Final Grade (need grade of 5 or more to be allowed to take the Final Exam)
- Final Exam (FE)
 - 2-3h examination, open books
 - partly on moodle (for FE grade < 7), optionally oral examination, for FE grade > 7 ; restrictions apply - TBD
 - algorithm traces, questions, algorithm design for specific problems, complexity analysis; open books
 - 50% of the Final Grade

Evaluation (4)

- Course Quizzes + MT (Q)
 - 20% of the Final Grade
- Hands on evaluation / laboratory assignments (L)
 - 30% in the Final Grade (need grade of 5 or more to be allowed to take the Final Exam)
- Final Exam (FE)
 - 50% of the Final Grade - 35% comprised of the mandatory Moodle examination (MM), and 15% of the optional oral examination (OO)
- bonuses apply for good seminar/course activity
- $FG = 0.2*Q + 0.3*L + (0.35*MM + 0.15*OO)FE + \text{Bonus}$
- starting last year, seminar activity presence is **MANDATORY!** (ECTS guide)



What to expect

- *NOT* a Programming course
- *NOT* a Data Structures course
- *It IS* a course on Fundamental Algorithms, learn (to):
 - *evaluate* algorithm *performance*
 - *compare performance* of different algorithms
 - *design efficient* and *optimal* algorithms
 - *identify a solution* to a given problem
 - *specific efficient algorithms* on *fundamental problems*