CS-101

Advanced information, computation, communication I

Käser Tanja

Cursus	Sem.	Type
Communication systems	BA1	Obl.
Computer science	BA1	Obl.
Cyber security minor	H	Opt.

Language of teaching	English
Coefficient	0
Session	Winter
Semester	Fall
Exam	Written
Workload	0h
Weeks	14
Hours	**6 weekly**
Courses	4 weekly
Exercises	2 weekly
Number of positions	

Remark

This course focuses on the foundational, discrete mathematics core of advanced computation.

Summary

Discrete mathematics is a discipline with applications to almost all areas of study. It provides a set of indispensable tools to computer science in particular. This course reviews (familiar) topics as diverse as mathematical reasoning, combinatorics, discrete structures & algorithmic thinking.

Content

I. Mathematical reasoning: propositional logic, propositional functions, quantifiers, rules of inference; this includes very basic logic circuits.
II. Sets and counting: cardinalities, inclusion/exclusion principle, sequences and summations.
III. Algorithms and complexity: basic algorithms, computational complexity, big-O notation and variants, countability.
IV. Number representations such as binary and hexadecimal and (postponed to 2nd semester) basic number theory: modular arithmetic, integer division, prime numbers, hash functions, pseudorandom number generation; applications.
V. Induction and recursion: mathematical induction, recursive definitions and algorithms.
VI. Basic combinatorial analysis: permutations, binomial theorem, basic generating functions.
VII. Basic probability: events, independence, random variables, Bayes' theorem.
VIII. Structure of sets: relations, equivalence relations, power set.
IX. (time permitting) Elementary graph theory: graphs, Euler and Hamilton paths, Dijkstra's algorithm, spanning trees.

Keywords

Propositional logic, counting, complexity, big-O, number representations, sets, matrices, modular arithmetic, induction, basic probabilities, Bayes theorem, combinatorial analysis, recurrences, generating functions, countability, graph theory.

Learning Outcomes

By the end of the course, the student must be able to:

- Recognize if there is a mistake in a (simple) proof
- Apply general problem-solving techniques
- Recognize the mathematical structures present in applications
- Apply simple recursion and use it to design recursive algorithms
- Apply the tools studied in class to solve problems
- Demonstrate familiarity with mathematical reasoning
- Solve linear recurrences and use generating functions
- Argue about (un)countability

- Formulate complete, clear mathematical proofs

Transversal skills

- Assess one's own level of skill acquisition, and plan their on-going learning goals.
- Continue to work through difficulties or initial failure to find optimal solutions.
- Demonstrate the capacity for critical thinking

Teaching methods

Ex cathedra lectures

Expected student activities

Studying the book, test your understanding by making the exercises, ask questions

Assessment methods

Continuous evaluations 10% and final exam 90%

Supervision

Office hours	No
Assistants	Yes
Forum	No
Others	Additional Q&A sessions will take place on Tuesdays from 17:15-18:30 in INM 10 (starting in the second week of the semester)

Resources

Bibliography

"Discrete Mathematics and Its Applications", Kenneth H. Rosen, 8th ed, McGraw-Hill 2019. (You should be able to find the pdf on the web.)

Ressources en bibliothèque

- Discrete mathematics and its applications / Rosen

Websites

- http://will be provided later, if any

Moodle Link

- https://go.epfl.ch/CS-101