



National University of Engineering (UNI)
School of Computer Science
Syllabus 2027-I

1. COURSE

CS311. Algorithms for Complex Problems (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: CS311. Algorithms for Complex Problems
2.2 Semester	: 6 th Semester.
2.3 Credits	: 3
2.4 Hours	: 2 HT; 2 HP;
2.5 Duration of the period	: 16 weeks
2.6 Type of course	: Mandatory
2.7 Learning modality	: Face to face
2.8 Prerequisites	: CS212. Analysis and Design of Algorithms. (5 th Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Competitive Programming trains students to solve complex algorithmic problems under time and resource constraints. This course bridges the gap between theoretical algorithm design and high-performance implementation. It covers advanced techniques in dynamic programming, graph theory, and mathematics, fostering the analytical thinking required for top-tier software engineering interviews and international programming contests like the ICPC.

5. GOALS

- Master advanced competitive programming techniques and paradigms.
- Develop the ability to quickly analyze the complexity of potential solutions.
- Implement efficient data structures for range queries and dynamic updates.
- Apply specialized mathematical and geometric algorithms to solve contests problems.
- Optimize code for execution speed and memory usage in a competitive environment.

6. COMPETENCES

AG-C08) Problem Analysis: Identifies, formulates, and analyzes complex computing problems. (Usage)

AG-C12) Uses systemic approaches to select, develop, apply, integrate, and manage secure computing technologies to achieve user objectives. (Usage)

7. TOPICS

Unit 1: Algorithmic Strategies (12 hours)	
Competences Expected: AG-C08,AG-C12	
Topics	Learning Outcomes
• Complete Search: Recursive Backtracking and Bit-masking. • Greedy Strategies: Choice property and optimal sub-structure. • Divide and Conquer: Binary Search on the answer and Ternary Search. • Dynamic Programming: State reduction, Digit DP, and DP on trees.	• Apply search pruning techniques to solve NP-hard problems in small constraints [Usar] • Identify the most efficient paradigm for a given problem statement [Evaluar]
Readings : [Halim2020], [Laaksonen2020]	

Unit 2: Foundational Algorithms (12 hours)	
Competences Expected: AG-C08,AG-C12	
Topics	Learning Outcomes
• Fenwick Trees (Binary Indexed Trees). • Segment Trees: Lazy Propagation and Persistent Segment Trees. • Square Root Decomposition and Mo's Algorithm. • Sparse Tables for RMQ.	• Implement range query structures for real-time data updates [Evaluar] • Analyze the trade-offs between different range query data structures [Usar]
Readings : [Halim2020], [Skiena2020]	

Unit 3: Foundational Algorithms (12 hours)	
Competences Expected: AG-C08,AG-C12	
Topics	Learning Outcomes
• Network Flow: Max Flow Min Cut (Dinic's and Edmonds-Karp). • Strongly Connected Components (Tarjan's and Kosaraju's). • Bipartite Matching and Heavy-Light Decomposition. • Centroid Decomposition on trees.	• For each algorithm in this unit explain step-by-step how the algorithm operates [Explain] • For each algorithmic approach (e.g., sorting) in this unit apply a prototypical example of the approach (e.g., merge sort) [Apply] • Given requirements for a problem, develop multiple solutions using various data structures and algorithms. Subsequently, evaluate the suitability, strengths, and weaknesses selecting an approach that best satisfies the requirements [Create] • Explain factors beyond computational efficiency that influence the choice of algorithms, such as programming time, maintainability, and the use of application-specific patterns in the input data [Explain] • For each of the algorithms and algorithmic approaches in the KA Core topics: – Explain a prototypical example of the algorithm, and – Explain step-by-step how the algorithm operates. [Explain]
Readings : [Laaksonen2020], [Halim2020]	

Unit 4: Foundational Algorithms (12 hours)	
Competences Expected: AG-C08,AG-C12	
Topics	Learning Outcomes
• Number Theory: Sieve of Eratosthenes, Extended Euclidean, and Modular Inverse. • Combinatorics: Inclusion-Exclusion and Burnside's Lemma. • Computational Geometry: Cross product, Convex Hull (Monotone Chain), and Line intersection. • Fast Fourier Transform (FFT) for polynomial multiplication.	• Solve combinatorial problems using modular arithmetic [Usar] • Implement geometric primitives for spatial relationship analysis [Evaluar]
Readings : [Skiena2020], [Laaksonen2020]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. EVALUATION SYSTEM

Theory Sessions:

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:

In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:

Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of our virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

The final grade NF depends on the following grades:

- PE : Partial Exam.
- FE : Final Exam.
- P : Permanent Evaluation (Average of practical work and graded assignments).

The final grade NF is calculated as follows:

$$NF = \frac{PE + FE + P}{3}$$

10. BASIC BIBLIOGRAPHY