



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2026-I

## 1. COURSE

CS311. Competitive Programming (Mandatory)

## 2. GENERAL INFORMATION

|                            |                                                                   |
|----------------------------|-------------------------------------------------------------------|
| 2.1 Course                 | : CS311. Competitive Programming                                  |
| 2.2 Semester               | : 6 <sup>th</sup> 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 <sup>th</sup> 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

| <b>Unit 1: Estrategias Algorítmicas (12 hours)</b>                                                                                                                                                                                                                                                                                               |                                                                                                                                                                                                                                   |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Competences Expected: AG-C08,AG-C12</b>                                                                                                                                                                                                                                                                                                       |                                                                                                                                                                                                                                   |
| Topics                                                                                                                                                                                                                                                                                                                                           | Learning Outcomes                                                                                                                                                                                                                 |
| <ul style="list-style-type: none"> <li>• Complete Search: Recursive Backtracking and Bit-masking.</li> <li>• Greedy Strategies: Choice property and optimal sub-structure.</li> <li>• Divide and Conquer: Binary Search on the answer and Ternary Search.</li> <li>• Dynamic Programming: State reduction, Digit DP, and DP on trees.</li> </ul> | <ul style="list-style-type: none"> <li>• Apply search pruning techniques to solve NP-hard problems in small constraints [Usar]</li> <li>• Identify the most efficient paradigm for a given problem statement [Evaluar]</li> </ul> |
| <b>Readings :</b> [HH20], [Laa20]                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                   |

| <b>Unit 2: Algoritmos Fundamentales (12 hours)</b>                                                                                                                                                                                                            |                                                                                                                                                                                                                  |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Competences Expected: AG-C08,AG-C12</b>                                                                                                                                                                                                                    |                                                                                                                                                                                                                  |
| Topics                                                                                                                                                                                                                                                        | Learning Outcomes                                                                                                                                                                                                |
| <ul style="list-style-type: none"> <li>• Fenwick Trees (Binary Indexed Trees).</li> <li>• Segment Trees: Lazy Propagation and Persistent Segment Trees.</li> <li>• Square Root Decomposition and Mo's Algorithm.</li> <li>• Sparse Tables for RMQ.</li> </ul> | <ul style="list-style-type: none"> <li>• Implement range query structures for real-time data updates [Evaluar]</li> <li>• Analyze the trade-offs between different range query data structures [Usar]</li> </ul> |
| <b>Readings :</b> [HH20], [Ski20]                                                                                                                                                                                                                             |                                                                                                                                                                                                                  |

| Unit 3: Algoritmos Fundamentales (12 hours)                                                                                                                                                                                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Competences Expected: AG-C08,AG-C12                                                                                                                                                                                                                                                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Topics                                                                                                                                                                                                                                                                                      | Learning Outcomes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <ul style="list-style-type: none"> <li>• Network Flow: Max Flow Min Cut (Dinic's and Edmonds-Karp).</li> <li>• Strongly Connected Components (Tarjan's and Kosaraju's).</li> <li>• Bipartite Matching and Heavy-Light Decomposition.</li> <li>• Centroid Decomposition on trees.</li> </ul> | <ul style="list-style-type: none"> <li>• Para cada algoritmo en esta unidad explicar paso a paso cómo opera el algoritmo [Explicar]</li> <li>• Para cada enfoque algorítmico (por ejemplo, ordenamiento) en esta unidad aplicar un ejemplo prototípico del enfoque (por ejemplo, ordenamiento por mezcla) [Aplicar]</li> <li>• Dados los requisitos para un problema, desarrollar múltiples soluciones usando varias estructuras de datos y algoritmos. Posteriormente, evaluar la idoneidad, fortalezas y debilidades seleccionando un enfoque que satisfaga mejor los requisitos [Crear]</li> <li>• Explicar factores más allá de la eficiencia computacional que influyen en la elección de algoritmos, como el tiempo de programación, la mantenibilidad y el uso de patrones específicos de la aplicación en los datos de entrada [Explicar]</li> <li>• Para cada uno de los algoritmos y enfoques algorítmicos en los temas del Núcleo KA: <ul style="list-style-type: none"> <li>– Explicar un ejemplo prototípico del algoritmo, y</li> <li>– Explicar paso a paso cómo opera el algoritmo.</li> </ul> [Explicar] </li> </ul> |
| Readings : [Laa20], [HH20]                                                                                                                                                                                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |

| Unit 4: Algoritmos Fundamentales (12 hours)                                                                                                                                                                                                                                                                                                                                          |                                                                                                                                                                                                        |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Competences Expected: AG-C08,AG-C12                                                                                                                                                                                                                                                                                                                                                  |                                                                                                                                                                                                        |
| Topics                                                                                                                                                                                                                                                                                                                                                                               | Learning Outcomes                                                                                                                                                                                      |
| <ul style="list-style-type: none"> <li>• Number Theory: Sieve of Eratosthenes, Extended Euclidean, and Modular Inverse.</li> <li>• Combinatorics: Inclusion-Exclusion and Burnside's Lemma.</li> <li>• Computational Geometry: Cross product, Convex Hull (Monotone Chain), and Line intersection.</li> <li>• Fast Fourier Transform (FFT) for polynomial multiplication.</li> </ul> | <ul style="list-style-type: none"> <li>• Solve combinatorial problems using modular arithmetic [Usar]</li> <li>• Implement geometric primitives for spatial relationship analysis [Evaluar]</li> </ul> |
| Readings : [Ski20], [Laa20]                                                                                                                                                                                                                                                                                                                                                          |                                                                                                                                                                                                        |

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BASIC BIBLIOGRAPHY

- [HH20] Steven Halim and Felix Halim. *Competitive Programming 4: The Lower Bound of Programming Contests*. Lulu, 2020.
- [Laa20] Antti Laaksonen. *Competitive Programmer's Handbook*. Draft, 2020.
- [Ski20] Steven S. Skiena. *The Algorithm Design Manual*. 3rd. Springer, 2020.