National University of the Altiplano (UNA)
School of Computer Science
Syllabus 2024-II

1. COURSE

CS311. Competitive Programming (Mandatory)

2. GENERAL INFORMATION

2.1 Course	:	CS311. Competitive Programming
2.2 Semester	:	6^{th} Semester.
2.3 Credits	:	4
2.4 Horas	:	2 HT; 4 HP;
2.5 Duration of the period	:	16 weeks
2.6 Type of course	:	Mandatory
2.7 Learning modality	:	Face to face
2.8 Prerrequisites	:	CS212. Analysis and Design of Algorithms. (5^{th} Sem) CS212. Analysis and Design of Algorithms. (5^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Competitive Programming combines problem-solving challenges with the fun of competing with others. It teaches participants to think faster and develop problem-solving skills that are in high demand in the industry. This course will teach you to solve algorithmic problems quickly by combining theory of algorithms and data structures with practice solving problems.

5. GOALS

- That the student uses techniques of data structures and complex algorithms..
- That the student apply the concepts learned for the application on a real problem.
- That the student investigate the possibility of creating a new algorithm and / or new technique to solve a real problem.

6. COMPETENCES

) ()

6) Apply computer science theory and software development fundamentals to produce computing-based solutions. ()

7. TOPICS

Unit 1: Introduction (20)	
Competences Expected:	
Topics	**Learning Outcomes**
• Introduction to Competetive Programming • Computacional model • Runtime and space complexity • Recurrence and recursion • Divide and conquer	• Identify and learn how to use the resources in the Random Access Machine (RAM) computacional model. [Usar] • Compute the runtime and space complexity for written algorithms. [Usar] • Compute the recurrence relations for recursive algorithms. [Usar] • Solve problems related to searching and sorting. [Usar] • Learning to select the right algorithms for divide-and-conquer problems. [Usar] • Design new algorithms for real-world problem solving.[Usar]
Readings : [**Cormen2009**], [**Steven09**], [**Kulikov09**], [**SkienaRevilla:PC:2003**], [**Laaksonen17**], [**aziz2012elements**]	

Unit 2: Data structure (20)	
Competences Expected:	
Topics	**Learning Outcomes**
• Arrays and strings problems • Linked lists problems • Stacks and queues problems • Trees problems • Hash tables problems • Heaps problems	• Recognize different data structures, their complexities, uses and restrictions.[Usar] • Identify the type of data structure appropriate to the resolution of the problem. [Usar] • Recognize types of problems associated with operations on data structures such as searching, inserting, deleting and updating.[Usar]
Readings : [**Cormen2009**], [**Steven09**], [**Kulikov09**], [**SkienaRevilla:PC:2003**], [**Laaksonen17**], [**aziz2012elements**]	

Unit 3: Algorithmic Design Paradigms (20)	
Competences Expected:	
Topics	**Learning Outcomes**
• Brute force • Divide and conquer • Backtracking • Greedy • Dynamic Programming	• Learning the different algorithhmic design paradigms.[Usar] • Learning to select the right algorithms for different problems applying different algorithhmic design paradigms.[Usar]
Readings : [**Cormen2009**], [**Steven09**], [**Kulikov09**], [**SkienaRevilla:PC:2003**], [**Laaksonen17**], [**aziz2012elements**]	

Unit 4: Graphs (20)	
Competences Expected:	
Topics	**Learning Outcomes**
• Graphs transversal • Graphs aplications • Shortest path • Networks and flows	• Identify problems classified as graph problems. [Usar] • Learn how to select the right algorithms for network problems (transversal, MST, shortest-path, network and flows). [Usar]
Readings : [**Cormen2009**], [**Steven09**], [**Kulikov09**], [**SkienaRevilla:PC:2003**], [**Laaksonen17**], [**aziz2012elements**]	

Unit 5: Advanced topics (20)	
Competences Expected:	
Topics	**Learning Outcomes**
• Number theory • Probabilities and combinations • String algorithms (tries, string hashing, z-algorithm) • Geometric algorithms	• Learning to select the right algorithms for problems in number theory and mathematics as they are important in competitive programming. [Usar] • Learning to select the right algorithms for problems about probabilities and combinations, strings and computational geometry. [Usar]
Readings : [**Cormen2009**], [**Steven09**], [**Kulikov09**], [**SkienaRevilla:PC:2003**], [**Laaksonen17**], [**aziz2012elements**]	

Unit 6: Domain specific problems (20)	
Competences Expected:	
Topics	**Learning Outcomes**
• Latency and throughput • Parallelism • Networks • Storage • High availability • Caching • Proxies • Load balancers • Key-value stores • Replicating and sharing • Leader election • Rate limiting • Logging and monitoring	• Learning to design systems for different domain-specific problems by applying knowledge about networks, distributed computing, high availability, storage and system architecture.[Usar]
Readings : [**Cormen2009**], [**Steven09**], [**Kulikov09**], [**SkienaRevilla:PC:2003**], [**Laaksonen17**], [**aziz2012elements**]	

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