



Universidad Nacional San Cristobal de Huamanga (UNSCH)

Programa Profesional de
Ciencia de la Computación
Sílabo 2024-II

1. CURSO

CS311. Competitive Programming (Mandatory)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CS311. Competitive Programming
2.2 Semestre	:	6 th Semester.
2.3 Créditos	:	4
2.4 horas	:	2 HT; 4 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Mandatory
2.7 Modalidad de aprendizaje	:	Face to face
2.8 Prerrequisitos	:	CS212. Analysis and Design of Algorithms. (5 th Sem) CS212. Analysis and Design of Algorithms. (5 th Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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. OBJETIVOS

- 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. RESULTADOS DEL ESTUDIANTE

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Usage)
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Usage)

7. TEMAS

Unidad 1: Introduction (20)	
Resultados esperados:	
Temas	Objetivos de Aprendizaje (<i>Learning Outcomes</i>)
• 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]
Lecturas : [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

Unidad 2: Data structure (20)	
Resultados esperados:	
Temas	Objetivos de Aprendizaje (<i>Learning Outcomes</i>)
• 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]
Lecturas : [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

Unidad 3: Algorithmic Design Paradigms (20)	
Resultados esperados:	
Temas	Objetivos de Aprendizaje (<i>Learning Outcomes</i>)
• Brute force • Divide and conquer • Backtracking • Greedy • Dynamic Programming	• Learning the different algorithmmic design paradigms.[Usar] • Learning to select the right algorithms for different problems applying different algorithmmic design paradigms.[Usar]
Lecturas : [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

Unidad 4: Graphs (20)	
Resultados esperados:	
Temas	Objetivos de Aprendizaje (<i>Learning Outcomes</i>)
• Graphs transversal • Graphs applications • 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]
Lecturas : [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

Unidad 5: Advanced topics (20)	
Resultados esperados:	
Temas	Objetivos de Aprendizaje (<i>Learning Outcomes</i>)
• 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]
Lecturas : [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

Unidad 6: Domain specific problems (20)	
Resultados esperados:	
Temas	Objetivos de Aprendizaje (<i>Learning Outcomes</i>)
• 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]
Lecturas : [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

***** EVALUATION MISSING *****

10. BIBLIOGRAFÍA BÁSICA

- [Mig03] Steve Skiena Miguel A. Revilla. *Programming Challenges: The Programming Contest Training Manual*. Springer, May 2003.
- [Cor+09] T. H. Cormen et al. *Introduction to Algorithms*. MIT Press, 2009.
- [ALP12] A. Aziz, T.H. Lee, and A. Prakash. *Elements of Programming Interviews: The Insiders' Guide*. ElementsOfProgrammingInterviews.com, 2012. URL: <https://books.google.com.pe/books?id=y6FLBQAAQBAJ>.
- [Hal13] Steven Halim. *Competitive Programming*. 3 rd. Lulu, 2013.
- [Laa17] Antti Laaksonen. *Guide to Competitive Programming: Learning and Improving Algorithms Through Contests*. Stringer, 2017.
- [Kul19] Alexander S. Kulikov. *Learning Algorithms Through Programming and Puzzle Solving*. Active Learning Technologies, 2019.