

## PROGRAMA DE CURSO PROGRAMACIÓN COMPETITIVA

### A. Antecedentes generales del curso:

Departamento	Ingeniería Matemática					
Nombre del curso	Programación Competitiva					
Nombre del curso en inglés	Competitive Programming					
Código	MA4205		Créditos	6		
Horas semanales	Docencia	1,5	Auxiliares	3	Trabajo personal	5,5
Carácter del curso	Obligatorio			Electivo	-----	
Requisitos	CC1002					

### B. Propósito del curso:

El propósito de este curso es introducir a alumnos del departamento de ingeniería matemática y plan común al mundo de la programación competitiva. La universidad de Chile ha sido un gran exponente en competencias de este tipo a nivel nacional desde el año 2012, pero ha ido reduciendo su nivel en los últimos años. Este curso pretende ser un primer paso en la rectificación de este hecho.

La programación competitiva es un deporte mental donde los participantes se enfrentan a un conjunto de problemas de naturaleza algorítmica. Generalmente se participa en equipos, y el equipo capaz de resolver la mayor cantidad de problemas en un tiempo acotado se declara ganador. En caso de empates en cantidad, se desempata según cual equipo utilizó menor tiempo en desarrollar sus soluciones.

En este curso el estudiante aprenderá conceptos básicos necesarios para poder participar en programación competitiva. Adicionalmente, se espera que el estudiante desarrolle habilidades que le permitan hacer uso de este entendimiento de forma práctica, aplicándolo a desafíos de programación. Se pone un especial énfasis en las habilidades de resolución de problemas y pensamiento crítico, entendido como la habilidad de construir sobre conocimiento ya adquirido para enfrentar problemas no antes vistos y crear nuevo conocimiento.

Se espera además que el estudiante actúe en las actividades que se le proponen de manera responsable y honesta, evitando por ejemplo el plagio, copia de sus pares u otras fuentes, así como

cualquier comportamiento que vaya en contra de sus pares, equipo docente o código de ética y reglamentos de la Escuela de la Facultad de Ciencias Físicas y Matemáticas.

En el curso esta competencia será evaluada de modo general cuyo detalle se describe en el apartado de "Estrategia de Evaluación" y no en base a resultados de aprendizaje.

### C. Resultados de aprendizaje:

#### Resultados de aprendizaje

**RA1:** Tiene un dominio suficiente de habilidades de programación, entendido como la capacidad de traducir ideas a códigos en un tiempo razonable.

**RA2:** Entiende conceptos útiles para el desarrollo de soluciones a problemas de programación competitiva, y es capaz de utilizar dichos conceptos para enfrentarse a problemas inéditos.

#### D. Unidades temáticas:

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
1	RA1	Introducción a Programación Competitiva	2 semanas
<b>Contenidos</b>		<b>Indicador de logro</b>	
1.1. Introducción al lenguaje de programación C++. 1.2. Noción de complejidad. 1.3. Algoritmo de búsqueda binaria.		El estudiante:  1. Tiene una noción básica de lo que es programación competitiva. 2. Puede utilizar herramientas básicas para resolver problemas de computación.	
Bibliografía de la unidad		[1]	

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
2	RA2	Programación Dinámica	2 semanas
<b>Contenidos</b>		<b>Indicador de logro</b>	
2.1. Programación Dinámica con Tabulación y Memoización. 2.2. Dividir y conquistar. 2.3. Recursión y Backtracking.		El estudiante:  1. Puede utilizar el paradigma de dividir y conquistar en la resolución de problemas. 2. Puede optimizar sus soluciones mediante el uso de programación dinámica.	
Bibliografía de la unidad		[1]	

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
3	RA2	Grafos	2 semanas
<b>Contenidos</b>		<b>Indicador de logro</b>	
3.1. Almacenamiento de grafos. 3.2. Algoritmos de recorrido en grafos: DFS y BFS. 3.3. Algoritmos de camino más corto con pesos no negativos: Floyd Warshall y Dijkstra. 3.4. Algoritmo de Kosaraju para encontrar componentes fuertemente conexas. 3.5. Problema del número de caminos distintos en grafos dirigidos acíclicos.		El estudiante:  1. Puede modelar problemas utilizando grafos. 2. Es capaz de implementar algoritmos sobre grafos.	
Bibliografía de la unidad		[1]	

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
4	RA2	Estructuras de Datos	2 semanas
Contenidos		Indicador de logro	
4.1. Segment Tree y variaciones. 4.2. Disjoint Set Union. 4.3. (Opcional) Heavy Light Decomposition.		El estudiante:  1. Puede utilizar las estructuras de datos vistas en la unidad como subrutinas en soluciones a problemas de programación competitiva.	
Bibliografía de la unidad		[1]	

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
5	RA2	Teoría de Números y Combinatoria	2 semanas
Contenidos		Indicador de logro	
1.1. Aritmética Modular. 1.2. Criba de Eratóstenes. 1.3. Exponenciación Binaria. 1.4. Inverso modular y teorema chino del resto. 1.5. Números especiales: Números de Catalán, Fibonacci, etc.		El estudiante:  1. Puede enfrentarse a problemas que requieran un conocimiento básico de teoría de números y conteo. 2. Es capaz de implementar algoritmos que apliquen estos conocimientos.	
Bibliografía de la unidad		[2]	

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
6	RA2	Redes y Flujo en redes	2 semanas
Contenidos		Indicador de logro	
6.1. Modelamiento de problemas combinatoriales como flujo en redes. 6.2. Algoritmo de Dinic para flujo máximo. 6.3. Teorema de Min-Cut Max-Flow. Obtención de corte mínimo mediante flujo máximo.		El estudiante:  1. Puede modelar problemas combinatoriales como problemas de flujo en redes. 2. puede utilizar el algoritmo de flujo máximo para desarrollar e implementar soluciones a problemas combinatoriales.	
Bibliografía de la unidad		[2]	

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
7	RA2	Geometría Computacional	2 semanas
Contenidos		Indicador de logro	
7.1. Primitivas: puntos y uso de producto punto y producto cruz. 7.2. Algoritmo de envoltura convexa. 7.3. Algoritmo para el cálculo de área de polígonos. 7.4. Teorema de Pick. 7.5. Algoritmo de ordenamiento angular.		El estudiante:  1. Puede desarrollar e implementar soluciones simples de problemas de geometría.	
Bibliografía de la unidad		[2]	

#### D. Estrategias de enseñanza:

La estrategia de enseñanza utilizada se compone de dos partes: clases expositivas con participación de los estudiantes, y sesiones de laboratorio donde el alumno aplica los conocimientos del curso de forma activa y guiada por profesores, mediante la resolución de problemas de programación.

El curso está compuesto por:

- Una clase teórica a la semana (1.5 hrs)
- Trabajo práctico en laboratorio (3.0 hrs)
- Trabajo individual o grupal de resolución de problemas (5.5 hrs).

#### E. Estrategias de evaluación:

El curso contempla distintas instancias de evaluación de proceso.

- Un control a mitad del semestre.
- Tareas bisemanales, trabajadas parcialmente en las sesiones prácticas.
- Examen final en formato presentación.

La ponderación de cada evaluación respetará siempre los reglamentos de la Escuela. En cada una de estas evaluaciones se evaluará la capacidad del estudiante para pensar creativamente

soluciones a problemas de programación. Esta evaluación se realiza de manera integral en la revisión de las evaluaciones y puede afectar un porcentaje de la calificación de cada una de ellas (como ejemplo, entre 1 y 5%).

#### F. Recursos bibliográficos:

##### **Bibliografía General:**

- (1) Competitive Programming 4 - Book 1, Steven & Felix Halim.
- (2) Competitive Programming 4 - Book 2, Steven & Felix Halim.

#### G. Datos generales sobre elaboración y vigencia del programa de curso:

Vigencia desde:	2024
Elaborado por:	Javier Marinkovic (DIM), Javier Oliva (DCC)
Validado por:	-
Revisado por:	-