

PROGRAMA DE CURSO

Código	Nombre			
CC4005	TALLER DE PROGRAMACION COMPETITIVA A			
Nombre en Inglés				
COMPETITIVE PROGRAMMING A				
SCT	Unidades Docentes	Horas de Discusión	Horas de Laboratorio	Horas de Trabajo Personal
6	10	2	5	3
Requisitos			Carácter del Curso	
CC1000 / CC1001			Electivo de Licenciatura	
Resultados de Aprendizaje				
<p>Al término del curso se espera que el estudiante conozca y aplique de manera efectiva un conjunto importante de las estructuras de datos y los algoritmos más usados en el ámbito de la programación competitiva, incluyendo Búsqueda Binaria, Programación Dinámica, Algoritmos Numéricos incluyendo Criba, Primalidad, Factorización, y Exponenciación Rápida, Algoritmos de Expansión de Grafos (BFS,DFS), Algoritmos sobre Árboles, Algoritmos de Caminos Mínimos, Componentes Fuertemente Conexas, Algoritmos de Flujo, y combinaciones de estos.</p> <p>El alumno será capaz de resolver en corto tiempo problemas de programación de alta complejidad sometido a requerimientos extremos de uso eficiente de tiempo y memoria computacional.</p> <p>El alumno adquirirá además las principales capacidades técnicas requeridas en la actualidad para trabajar de buena manera en empresas de alta tecnología e innovación (tipo Google, Facebook, Microsoft, Twitter) y estará capacitado para enfrentarse de manera exitosa a las típicas tareas exigidas en los procesos de reclutamiento de estas.</p> <p>Dado que gran parte del aprendizaje es autónomo, el alumno adquirirá capacidades de auto aprendizaje, investigación, y testing de estrategias de solución.</p> <p>Finalmente, el alumno aprenderá a optimizar recursos escasos en la resolución de problemas computacionales y a trabajar en equipos con miembros de distinta expertise para solucionar problemas complejos.</p>				

Metodología Docente	Evaluación General
<p>La metodología es del tipo “flipped-classroom” en casi la totalidad del curso, y compone:</p> <ul style="list-style-type: none"> - Aprendizaje autónomo mediante recursos on-line de tópicos específicos (en la forma de videos de clases, y lectura complementaria) - Trabajo práctico grupal en laboratorio: 5 horas semanales - Discusión presencial de tópicos y soluciones de problemas: 2 horas semanales. - Trabajo individual de resolución de problemas: 3 horas semanales. 	<p>La evaluación se basa en trabajo práctico en laboratorio, trabajo individual y participación.</p> <ul style="list-style-type: none"> - Cada laboratorio tiene una nota que depende de la cantidad de desafíos que el grupo de alumnos alcance a resolver durante las 5 horas. - Cada alumno deberá además exponer la solución de al menos uno de los problemas resueltos durante el semestre y deberá participar activamente en la discusión de problemas expuestos por sus compañeros. - Adicionalmente habrá tareas individuales temáticas que tienen entregas cada dos semanas. <p>La nota se calcula como:</p> <ul style="list-style-type: none"> - 50% trabajo en laboratorio, - 20% participación en discusiones, - 30% tareas individuales.

Unidades Temáticas

Número	Nombre de la Unidad	Duración en Semanas
1	Introducción a la Programación Competitiva	1
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
<ul style="list-style-type: none"> - Problemas de programación ad-hoc, de simulación, o de estructuras de datos simples como colas, listas, arreglos. - Análisis simple de complejidad de algoritmos y aplicación para estimar el tiempo de ejecución de un programa. 	<p>Al término de la unidad, el alumno conoce el formato de problemas competitivos y puede escribir programas para resolver los problemas básicos, analizando si su programa cumplirá con los requisitos de tiempo de ejecución y uso de memoria.</p>	<p>[CP3, cap. 1, 2] [PCh, cap. 1, 2]</p>

Número	Nombre de la Unidad	Duración en Semanas
2	Búsqueda Binaria en Programación Competitiva	2
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
<p>Búsqueda Binaria general para problemas de programación competitiva: dividir para conquistar, búsqueda binaria de la solución.</p>	<p>El alumno es capaz de identificar cuándo un problema de programación competitiva simple puede resolverse usando Búsqueda Binaria, y es capaz de implementar correctamente una solución que implique el uso de esta estrategia.</p>	<p>[CP3, cap. 3.3]</p>

Número	Nombre de la Unidad	Duración en Semanas
3	Programación Dinámica en Programación Competitiva	2
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
Uso práctico de programación dinámica, programación dinámica lineal, 2D, 3D, con estado basado en bitsets.	El alumno es capaz de identificar cuándo un problema de programación competitiva simple puede resolverse usando Programación Dinámica, y es capaz de implementar correctamente una solución que implique el uso de esta estrategia.	[CP3, cap. 3.5] [PCh, cap. 11]

Número	Nombre de la Unidad	Duración en Semanas
4	Algoritmos Numéricos en Programación Competitiva	2
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
Criba de Eratóstenes, Criba extendida, Función de Euler, Algoritmo de Euclides, Algoritmo Extendido de Euclides, Aritmética Modular, Exponenciación Rápida.	El alumno es capaz de implementar los principales algoritmos numéricos y utilizarlos para solucionar problemas de programación competitiva simples.	[CP3 cap. 5] [PCh cap. 7]

Número	Nombre de la Unidad	Duración en Semanas
5	Algoritmos Básicos de Árboles y Grafos en Programación Competitiva	2
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
Recorridos en amplitud (BFS) y profundidad (DFS), coloración de nodos y aristas, chequeo de ciclos, algoritmos básicos sobre árboles.	El alumno es capaz de identificar cuándo un problema de programación competitiva simple puede modelarse como un grafo y solucionarse recorriéndolo con BFS o DFS. Es capaz de implementar estas estrategias y utilizarlas eficientemente.	[CP3 cap. 4.2, 4.3] [PCh cap. 9, 10.2]

Número	Nombre de la Unidad	Duración en Semanas
6	Algoritmos de Caminos más Cortos en Programación Competitiva	2
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
Algoritmos de Dijkstra, Floyd-Warshall, y sus diversos usos en programación competitiva.	El alumno es capaz de identificar cuándo un problema de programación competitiva simple puede modelarse como un grafo y solucionarse con un algoritmo de caminos más cortos o modificaciones de éstos. Es capaz de implementar estos algoritmos.	[CP3 cap. 4.4, 4.5] [PCh cap. 10.3]

Número	Nombre de la Unidad	Duración en Semanas
7	Otros Algoritmos de Grafos en Programación Competitiva	2
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
Orden Topológico, Componentes Fuertemente Conexas, Flujo, Matching Bipartito.	El alumno es capaz de identificar cuándo un problema de programación competitiva simple puede modelarse como un grafo y solucionarse con diversos algoritmos de grafos. Es capaz de implementar estos algoritmos.	[CP3 cap. 4.6, 4.7.1, 4.2.9] [PCh cap. 10.4, 10.2]

Número	Nombre de la Unidad	Duración en Semanas
8	Problemas Compuestos y Tópicos Avanzados en Programación Competitiva	2
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
Estrategias que combinan las estrategias vistas en las unidades anteriores. Opcionalmente, estrategias avanzadas a elegir entre: Algoritmo de Mo, HeavyLight Decomposition, Transformada de Fourier Rápida, Optimización de Programación Dinámica, Autómatas de Sufijos, Wavelet Trees, etc.	El alumno es capaz de implementar estrategias complejas que combinen una o varias estrategias vistas durante el semestre, para solucionar problemas de programación competitiva.	[CP3, cap. 8-9]

Bibliografía

- Steven Halim, Felix Halim, Competitive Programming 3 [CP3]
- Steven Skiena, Miguel Revilla, Programming Challenges [PCh]

Vigencia desde:	Otoño 2017
Elaborado por:	Jorge Pérez Rojas