

PROGRAMA DE CURSO

Código	Nombre			
CC4006	TALLER DE PROGRAMACION COMPETITIVA B			
Nombre en Inglés				
COMPETITIVE PROGRAMMING B				
SCT	Créditos	Horas de Discusión	Horas de Laboratorio	Horas de Trabajo Personal
6	6	2	5	3
Requisitos			Carácter del Curso	
Sin requisitos.			Electivo de Licenciatura en Computación.	
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, Back Tracking, Fenwick Trees, Segment Trees, Rolling Hashing, Tries, Suffix Arrays, Algoritmos Geométricos 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	Búsqueda Exhaustiva en Programación Competitiva	2
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
Uso práctico de Búsqueda Exhaustiva (Back Tracking) en Programación Competitiva, incluyendo estrategias simples de poda.	El alumno es capaz de identificar cuándo un problema de programación competitiva simple puede resolverse usando Búsqueda Exhaustiva, y es capaz de implementar correctamente y eficientemente una solución que implique el uso de esta estrategia, decidiendo una buena forma de representar el espacio de búsqueda.	[CP3, cap. 3.2] [PCh, cap. 8]

Número	Nombre de la Unidad	Duración en Semanas
4	Árboles Indexados Binariamente (BITS, Fenwick Trees) en Programación Competitiva	2
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
Uso práctico de BITS y Fenwick Trees, uso directo, computación de frecuencias acumuladas, sumas parciales con actualización, Rank + Select.	El alumno es capaz de identificar cuándo un problema de programación competitiva simple requiere el uso de una estructura de tipo Fenwick Tree. Es capaz de implementar esta estructura y utilizarla eficientemente.	[CP3 cap. 2.4.4]

Número	Nombre de la Unidad	Duración en Semanas
5	Segment Trees en Programación Competitiva	2
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
Segment Trees y estructuras similares para cálculos de operaciones agregadas en presencia de actualizaciones, actualizaciones por rangos y Segment Tree con Lazy Update.	El alumno es capaz de identificar cuándo un problema de programación competitiva simple requiere el uso de una estructura de tipo Segment Tree o similares. Es capaz de implementar esta estructura y utilizarla eficientemente, incluyendo implementaciones simples de actualización Lazy de la misma.	[CP3 cap. 2.4.3]

Número	Nombre de la Unidad	Duración en Semanas
6	Algoritmos Sobre Strings en Programación Competitiva	2
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
Rolling Hashing, Tries, Suffix Arrays para diversas aplicaciones en programación competitiva.	El alumno es capaz de identificar cuándo un problema de programación competitiva simple puede ser resuelto usando conceptos de strings. Entiende las principales características de Rolling Hashing y Suffix Arrays su uso versátil. Entiende cuándo una estructura tipo Trie es necesaria. Es capaz de implementar estas estrategias y utilizarlas eficientemente.	[CP3 cap. 6] [PCh cap. 3]

Número	Nombre de la Unidad	Duración en Semanas
7	Algoritmos Geométricos en Programación Competitiva	2
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
Puntos, líneas, intersecciones, áreas, envoltura convexa, técnica de sweep-line.	El alumno es capaz de implementar estrategias simples de geometría para resolver problemas de programación competitiva.	[CP3 cap. 7] [PCh cap. 13]

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:	Primavera 2018
Elaborado por:	Jorge Pérez Rojas