

PROGRAMA DE CURSO

Código	Nombre			
CC4102	DISEÑO Y ANÁLISIS DE ALGORITMOS			
NOMBRE EN INGLÉS				
DESIGN AND ANALYSIS OF ALGORITHMS				
SCT	Unidades Docentes	Horas de Cátedra	Horas Docencia Auxiliar	Horas de Trabajo Personal
6	10	3	1.5	5.5
Requisitos			Carácter del Curso	
FI2003 Métodos Experimentales, CC3102 Teoría de la Computación, (MA3403/MA3401)/Autor			Obligatorio para Licenciatura en Computación.	
Resultados de Aprendizaje				
<p>Se espera que el alumno</p> <ul style="list-style-type: none"> Domine a un nivel elemental: el concepto de complejidad de un problema y técnicas para demostrarla, técnicas de diseño de algoritmos y estructuras de datos para memoria secundaria, análisis amortizado de algoritmos, algoritmos en línea, algoritmos aleatorizados y probabilísticos, algoritmos aproximados, y algoritmos paralelos. Conozca un conjunto significativo de algoritmos y estructuras de datos de mediana complejidad para solución de problemas fundamentales. 				

Metodología Docente	Evaluación General
<ul style="list-style-type: none"> Clases expositivas del profesor de cátedra, buscando la participación de los alumnos en pequeños problemas que se van proponiendo durante la exposición. Clases auxiliares dedicadas a explicar ejemplos más extensos, resolver ejercicios propuestos, y preparación pre y post controles. Exposición de las mejores tareas de los alumnos, como casos de estudio de implementación y experimentación. 	<p>Se realizan tres controles para evaluar si se han cumplido los objetivos.</p> <p>El primero evalúa las unidades 1 y 2, El segundo la unidad 3 El tercero la unidad 4.</p> <p>El examen evalúa todas las unidades, en particular la parte de la 4 que no llega a evaluarse completamente en el control.</p> <p><i>La Nota de Final se calcula de la siguiente manera: Promedio ponderado del examen (40%) Promedio de los controles (60%)</i></p> <p>Tareas (3)</p> <ul style="list-style-type: none"> Implementar algoritmos o estructuras de datos alternativos para resolver un cierto problema, comparando las soluciones básicas con las que se introducen en el curso <p><i>Controles y tareas se aprueban por separado y deben ser igual o superior a 4.0. La nota final es 2/3 de la nota de controles y 1/3 de la nota de tareas.</i></p>

Unidades Temáticas

Número	Nombre de la Unidad	Duración en Semanas
1	Conceptos básicos y complejidad	3
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
1. Repaso del proceso de diseño y análisis de un algoritmo. Metodología de experimentación. 2. Técnicas para demostrar cotas inferiores: adversario, teoría de la información, reducción. 3. Principales casos de estudio (o similares): caso promedio del quicksort, cota inferior para mínimo y máximo de un arreglo, cota inferior para búsqueda en un arreglo con distintas probabilidades de acceso.	Al término de la unidad el alumno: <ul style="list-style-type: none"> • Maneja nociones básicas de experimentación en algoritmos. • Comprenda el concepto de complejidad de un problema como cota inferior, y reconozca técnicas elementales para demostrar cotas inferiores. • Conoce algunos casos de estudio relevantes. 	[1] Cap 1-4. [3] Cap 2, 10. [4] Cap 1, 4. [5] Cap 2. [6] Cap 1-2, 10. [7] Cap 6. [9] Cap 8-9. [10] Cap 1,2.

Número	Nombre de la Unidad	Duración en Semanas
2	Algoritmos y Estructuras de Datos para Memoria Secundaria	3
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
1. Modelo de computación en memoria secundaria. Accesos secuenciales y aleatorios 2. Ordenamiento en memoria secundaria: Mergesort. Cota inferior. 3. Colas de prioridad en memoria secundaria. Cotas inferiores. 4. Diccionarios en memoria secundaria: árboles B, hashing lineal y extendible.	Al término de la unidad el alumno: <ul style="list-style-type: none"> • Comprende el modelo de costo de memoria secundaria. • Reconoce algoritmos y estructuras de datos básicos que son eficientes en memoria secundaria, y el análisis de su desempeño. 	[1] Cap 18.[5] Cap 4.7, 7.11. [7] Cap 13, 18. [9] Cap 11.[10] Cap 5.7, 6.3.

Número	Nombre de la Unidad	Duración en Semanas
3	Técnicas avanzadas de diseño y análisis de algoritmos	4
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
<p>1. Análisis amortizado de algoritmos y estructuras de datos: análisis completo, contabilidad de costos, función potencial.</p> <p>2. Uso de dominios discretos y finitos en el diseño de algoritmos.</p> <p>3. Algoritmos en línea. Competitividad.</p> <p>4. Principales casos de estudio (o similares): estructuras para union-find, colas binomiales, splay trees, búsqueda por interpolación, radix sort, árboles de van Emde Boas, árboles de sufijos, técnica de los cuatro rusos, paginamiento, búsqueda no acotada.</p>	<p>Al término de la unidad el alumno:</p> <ul style="list-style-type: none"> Comprende las técnicas de algoritmos de costo amortizado, uso de finitud, y algoritmos competitivos. Diseña y analiza algoritmos y estructuras de datos basados en estos principios. Conoce algunos casos de estudio relevantes. 	<p>[1] Cap 8, 17, 19, 21. [2] Cap 4. [3] Cap 6. [5] Cap 4-6, 8, 11. [7] Cap 17.[8] Cap 9. [9] Cap 5. [10] Cap 3.3.</p>

Número	Nombre de la Unidad	Duración en Semanas
4	Algoritmos no convencionales	5
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
<p>1. Algoritmos aleatorizados y probabilísticos. Ejemplos en que no hay otra alternativa. Relación con la NP-completitud.</p> <p>2. Algoritmos tipo Monte Carlo y Las Vegas.</p> <p>3. Aleatorización de la entrada. Independencia de la distribución de la entrada. Estructuras de datos</p>	<p>Al término de la unidad el alumno:</p> <ul style="list-style-type: none"> Comprende el concepto de algoritmos aleatorizados, probabilísticos, aproximados, y paralelos, y cuándo son relevantes. - diseña y analiza algoritmos de estos tipos. Conoce algunos casos de estudio relevantes. 	<p>[1] Cap 5, 31.8, 32.2, 35. [3] Cap 6, 11, 12. [4] Cap 6. [5] Cap 10.[6] Cap 8. [7] Cap 35, 40, 44. [8] Cap 1, 7, 8, 12, 14. [10] Cap 4, 12.</p>

<p>aleatorizadas.</p> <p>4. Solución de problemas NP-completos: búsqueda exhaustiva. Concepto de algoritmos aproximados.</p> <p>5. Nociones de aproximabilidad. Problemas que son o no aproximables.</p> <p>6. Algoritmos paralelos y distribuidos. Medidas de complejidad. Técnicas de diseño.</p> <p>7. Principales casos de estudio (o similares): primalidad, Karp-Rabin para búsqueda en strings, número mayoritario, árboles binarios de búsqueda aleatorizados, quicksort, hashing universal y perfecto, aproximaciones para recubrimiento de vértices, vendedor viajero, mochila. Ordenamiento paralelo, parallel-prefix.</p>		
---	--	--

Bibliografía	
[1]	T. Cormen, C. Leiserson, R. Rivest, C. Stein. Introduction to Algorithms, 2 nd edition. MIT Press, 2001.
[2]	A. Aho, J. Hopcroft, J. Ullman. The Design and Analysis of Computer Algorithms. Addison-Wesley, 1974.
[3]	U. Manber. Introduction to Algorithms. Addison-Wesley, 1989.
[4]	G. Rawlins. Compared to what? Computer Science Press, 1992.
[5]	M. Weiss. Data Structures and Algorithm Analysis, 2 nd edition. Benjamin Cummings, 1995.
[6]	G. Brassard, P. Bratley. Algorithmics. Theory and Practice. Prentice-Hall, 1988.
[7]	R. Sedgewick. Algorithms in C++. Addison-Wesley, 1992.
[8]	R. Motwani, P. Raghavan. Randomized Algorithms. Cambridge, 1995.
[9]	A. Aho, J. Hopcroft, J. Ullman. Data Structures and Algorithms. Addison-Wesley, 1983.
[10]	K. Mehlhorn and P. Sanders. Algorithms and Data Structures. Springer, 2008.

Vigencia desde:	2010
Elaborado por:	Gonzalo Navarro / Jeremy Barbay
Revisado por:	ADD (septiembre de 2010)