

PROGRAMA DE CURSO

Código	Nombre			
CC74D	Análisis Fino de Algoritmos y Estructuras de Datos			
Nombre en Inglés				
Fine Analysis of Algorithms and Data Structures				
SCT	Unidades Docentes	Horas de Cátedra	Horas Docencia Auxiliar	Horas de Trabajo Personal
6	10	3	0	7
Requisitos			Carácter del Curso	
CC3001			Electivo	
Resultados de Aprendizaje				
<p>El alumno que ha aprobado este curso:</p> <ul style="list-style-type: none"> - Sabe explicar los conceptos vistos en el curso CC3001; - Sabe analizar algoritmos o estructuras de datos en el peor caso sobre clases de instancias arbitrariamente finas; - Sabe diseñar nuevas estructuras de datos comprimidas y nuevos algoritmos adaptativos en función de un análisis dado; - Tiene asegurados y reforzados los conocimientos requeridos para tomar el curso CC4102 "Diseño y Análisis de Algoritmos". 				

Metodología Docente	Evaluación General
<ul style="list-style-type: none"> -Dos charlas de introducción en español -Presentación en español de artículos científicos en inglés. -Presentación de trabajos de Investigación o de Pedagogía. 	<ul style="list-style-type: none"> -Preguntas de Lecturas (5%) -Presentaciones (40%) -Evaluación de Presentación (10%) -Preguntas de Concepto (5%) -Trabajo de Investigación o de Pedagogía (40%)

Unidades Temáticas

Número	Nombre de la Unidad	Duración en Semanas
1	Principios de Análisis	1,5
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
Hanoi Tower -> Disk Pile Problem	<ul style="list-style-type: none"> - Peor caso por n fijado - Peor caso por n,delta fijado - Peor caso por delta,n_1, ...,n_delta fijado 	[FUN 2016]

Número	Nombre de la Unidad	Duración en Semanas
2	Operaciones sobre Conjuntos Ordenados	2,5
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
<ol style="list-style-type: none"> 1. Binary Search -> Doubling Search 2. Binary Encoding -> Gamma Coding 3. (2,4) Search Trees -> Finger Search Tree. 4. Splay Trees 5. Sorted Merge -> Adaptive Sorted Merge. 6. Sorted Intersection -> Adaptive Insertion. 	<ul style="list-style-type: none"> - Búsqueda ordenada y Codificación de Enteros - Ordenamiento en el modelo de comparación y Compresión de permutaciones - Uso recursivo de la adaptividad - Casos de estudio 	[SODA 2000, CACM 1980, ACMCS 1992, TC 1985, DAM 1995, TALG 2006]

Número	Nombre de la Unidad	Duración en Semanas
3	Ordenamiento	8
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
1. Insertion Sort -> Local Insertion Sort 2. Heap -> Adaptive Priority queues 3. Heap sort -> Smooth Sort 4. Sorted Merge -> Adaptive Sorted Merge. 5. Merge Sort -> Adaptive MergeSort 6. Huffman's algorithm -> Instance Optimal Prefix Free Codes.	- Reducción adaptativa - Adaptividad Optimal - Optimalidad sobre Instancias (en profundidad)	[CPM 2016, TCS 2007, SPDS 2001, TC 1985, DAM 1995]

Número	Nombre de la Unidad	Duración en Semanas
4	Geometría Computacional Adaptativa	3
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
1. Convex Hull -> Output Sensitive Convex Hull 2. Dominating Set -> Instance Optimal DS 3. MCS Trees -> MCS Splay Trees 4. Optimal Boxes -> Adaptive Optimal Boxes.	- Sensitividad al tamaño del output - Optimalidad sobre Instancias (en profundidad)	[JACM 2017, DCG 1996, JC 1986,

Bibliografía
<ul style="list-style-type: none"> - P. Afshani, J.Barbay, T. Chan. Instance-Optimal Geometric Algorithms. JACM 2017. - J. Barbay. Optimal Prefix Free Codes with Partial Sorting. CPM 2016. - J. Barbay. Selenite Towers move faster than Hanoi Towers, but still require exponential time. FUN 2016.

- J. Barbay, A. Golynski, J. I. Munro, and S. S. Rao. Adaptive searching in succinctly encoded binary relations and tree-structured documents. *ELSEVIER Theoretical Computer Science (TCS)* , October 2007.
- J. Barbay and C. Kenyon. Alternation and redundancy analysis of the intersection problem. *ACM Transactions on Algorithms (TALG)* , May 2006.
- T. M. Chan. Output-sensitive results on convex hulls, extreme points, and related problems. *Discrete & Computational Geometry* , 16:369--387, 1996.
- C. Cool and D. Kim. Best sorting algorithm for nearly sorted lists. *Communication of ACM* , 23:620--624, 1980.
- E. D. Demaine, A. Lopez-Ortiz, and J. I. Munro. Adaptive set intersections, unions, and differences. In *Proceedings of the 11th ACM-SIAM Symposium on Discrete Algorithms (SODA)* , pages 743--752, 2000.
- V. Estivill-Castro and D. Wood. A survey of adaptive sorting algorithms. *ACM Computing Surveys* , 24(4):441--476, 1992.
- R. Fagin, A. Lotem, and M. Naor. Optimal aggregation algorithms for middleware. In *Symposium on Principles of Database Systems* , 2001.
- D. G. Kirkpatrick and R. Seidel. The ultimate planar convex hull algorithm?. *SIAM J. Comput.* , 1986. 15(1):287--299.
- H. Mannila. Measures of presortedness and optimal sorting algorithms. In *IEEE Trans. Comput.* , volume 34, pages 318--325, 1985.
- O. Petersson and A. Moffat. A framework for adaptive sorting. *Discrete Applied Mathematics* , 59:153--179, 1995.

Vigencia desde:	Marzo 2017
Elaborado por:	Jérémy Barbay