GUIA DOCENTE DE LA ASIGNATURA

Design and Analysis of Algorithms

Curso 2014- 2015

MÓDULO	MATERIA	CURSO	SEMESTRE	CRÉDITOS	TIPO
	Design and Analysis of Algorithms		1º	5	Mandatory

PROFESORES*	DIRECCIÓN COMPLETA DE CONTACTO PARA TUTORÍAS (Dirección postal, teléfono, correo electrónico, etc.)
Jose Manuel Benitez (University of Granada)	http://directorio.ugr.es/static/Personal/*/j.m.benitez_at_decsai.ugr.es
	HORARIO DE TUTORÍAS*

MASTER EN EL QUE SE IMPARTE	
Máster EM Color in Informatics and Media Technology CIMET	

PRERREQUISITOS Y/O RECOMENDACIONES (si procede)

Sufficient knowledge of Data structures and algorithms. Image analysis and processing course (1st semester).

BREVE DESCRIPCIÓN DE CONTENIDOS (SEGÚN MEMORIA DE VERIFICACIÓN DEL GRADO)

Specification of the concept of algorithm and analysis of its computational complexity. Design principles of algorithms and their application to computing problems. Topics include theory of NP-completeness, analysis techniques, and the main design principles such as divide-and-conquer, dynamic programming, branch-and-bound. Heap data structure and advanced binary search trees are also studied. Approximation, randomized and optimization techniques are considered for finding suboptimal solutions to NP-complete problems. These include local search, genetic algorithms and swarm intelligence.

COMPETENCIAS GENERALES Y ESPECÍFICAS

* Consulte posible actualización en Acceso Identificado > Aplicaciones > Ordenación Docente.

On completion of this course the students will be able to: - Design algorithms for difficult problems. - Analyze and understand their complexity. - Being able to implement the algorithms in practice

OBJETIVOS (EXPRESADOS COMO RESULTADOS ESPERABLES DE LA ENSEÑANZA)

TEMARIO DETALLADO DE LA ASIGNATURA

Topics to be taught (may be modified):

- Introduction to complexity theory. Why is complexity an important topic? What are the elements that influence the fact that a program solves in an acceptable mount of time a problem? How complexity is computed: recurrences, asymptotics, Concrete complexity
- Greediness. Characterisation. Examples: minimum spanning trees, other graph algorithms
- Divide and conquer. Characterisation. Examples to be added. Many algorithms correspond to trees.
- Dynamic Programming 1 (due to the importance of this family of algorithms in image processing and pattern recognition, 2 lectures). Examples: HMM algorithms (Forward, Backward, Viterbi), edit distance algorithms, optimal allocation of constrained resource, optimal partition of data sequence, shortest path in graphs (Trellis and directed acylic graphs)
- Dynamic Programming 2
- Organising the data 1: once the best possible algorithm is found (?), what else can we do? We can aim to find an alternative representation of the data, in which case (but usually at a price) we can find new, faster algorithms. Examples : Huffman encoding, red/black trees, heaps, hashing
- Organising the data 2:
- Proving that a problem is intractable: NP-hard problems. NP completeness, NP-hardness. Reduction techniques. Classes P and NP, polynomial certificate, reductions
- Visiting different NP-complete problems. Giving different examples of reductions and therefore of NP-complete problems: Graphs (colouring, dominating set, clique), strings (longest common subsequence), arrays…
- Randomisation as a means to get results faster with a possible error. Monte Carlo and Las Vegas algorithms. Examples.
- Combinatorial optimisation: accepting not to find the best solution but hoping for a good one. Gradient descent, Tabu search, genetic algorithms, Ant colonies…

Practical Laboratory Sessions: Note that the idea is not to teach programming language. Each student should be allowed to use the programming language he/she prefers (provided the language can handle usual data structures. Examples can be C++, C, Java, CAML,… Typically the sessions could involve visiting several pattern recognition problems over different paradigms and compare the methods.

BIBLIOGRAFÍA

Reference book:

- Jon Kleinberg and Eva Tardos, “Algorithm Design“, Pearson International Edition, 2006.

Additional textbooks and lecture materials:

- T. Cormen, C. Leiserson, and R. Rivest and C.Stein: *Introduction to Algorithms*, MIT Press, 2nd edition 2001.
- Levitin: *The design and analysis of algorithm*, Addison Wesley, 2007.
- P. Fränti, *Introduction to Combinatoric Optimization Techniques*, Lecture Notes, 2004

ENLACES RECOMENDADOS

METODOLOGÍA DOCENTE

Teaching methods: Lectures and lab classes, and homework exercises.

EVALUACIÓN (INSTRUMENTOS DE EVALUACIÓN, CRITERIOS DE EVALUACIÓN Y PORCENTAJE SOBRE LA CALIFICACIÓN FINAL, ETC.)

INFORMACIÓN ADICIONAL