SUBJECT TEACHING GUIDE

G682 - Algorithm Design

Degree in Computer Systems Engineering

Academic year 2021-2022

1. IDENTIFYING DATA					
Degree	Degree in Computer Systems Engineering			Type and Year	Optional. Year 4
Faculty	Faculty of Sciences				
Discipline	Subject Area: Computing Mention in Computing				
Course unit title and code	G682 - Algorithm Design				
Number of ECTS credits allocated	6	Term		Semester based (1)	
Web					
Language of instruction	Spanish	English Friendly	No	Mode of delivery	Face-to-face

Department	DPTO. MATEMATICAS, ESTADISTICA Y COMPUTACION
Name of lecturer	DOMINGO GOMEZ PEREZ
E-mail	domingo.gomez@unican.es
Office	Facultad de Ciencias. Planta: + 3. DESPACHO DOMINGO GOMEZ PEREZ (3005)
Other lecturers	JOSE LUIS MONTAÑA ARNAIZ SANTOS BRINGAS TEJERO

3.1 LEARNING OUTCOMES
- Deepening the study of techniques regarding the efficiency of algorithms. Reasoning about the correctness and efficiency. Understand classical algorithms for fundamental problems. Knowing how to identify the most relevant components of a problem and select the most appropriate algorithmic technique. Being able to choose the most appropriate data types to improve on the efficiency of an algorithmic solution.
- Understand and know how to use algorithm design strategies. For each of the following techniques: greedy algorithms, divide and conquer, backtracking, dynamic programming and branch and bound : the general scheme , the efficiency analysis in the worst case, and applying to some classical problems.
- Know how to use advanced data structures : tries, mf- sets, open and closed dispersion.
- Understand the concepts of probabilistic algorithm and different kinds of complexity associated with them . Know and understand the concept of approximate algorithm and probably approximately correct algorithm.
- Know and understand the concept of heuristic algorithm and metaheuristic technique. Knowing when a problem must be attacked by heuristics .

4. OBJECTIVES
To Know the layout of the greedy algorithms , to identify when and how to apply , meet the most common techniques demonstration of the correctness of these algorithms , and become familiar with some fundamental , such as Dijkstra's algorithm , Kruskal and Prim greedy algorithms .
Knowing the dynamic programming scheme , identify when and how you can apply and become familiar with some fundamental dynamic programming algorithms , for example , Floyd algorithm or calculation of the edit distance
Knowing the basic problem of calculating optimal flows on networks , familiar with a basic algorithm (Ford - Fulkerson) , understand the theorem Maxflow - mincut , recognize when a problem can be formulated in terms of a flow problem
Knowing the formulation of linear programming (LP) , to understand how many computational problems can be formulated in terms of LP , specifically the problems of flows in networks , understand the principles of algorithm simplex , understand the fundamental relationship between a problem LP (primal) and its dual
Understand the importance of randomization in the design of algorithms and data structures, familiarize yourself with some basic techniques of probabilistic analysis necessary to study the efficiency of algorithms aleatoritados and become familiar with some classic examples as randomized quicksort , the skip list , the primality test or Rabin search algorithm Karp - Rabin patterns
Knowing some specific computational problems in fields as diverse as searching document databases , databases of protein and genomic data, geographic information systems , information retrieval based on content , data compression , etc. and learn some advanced data structures to respond to these needs
Be familiar with the use of algorithmic design principles for the design of data structures and know some essential techniques to obtain these implementations to ensure maximum efficiency it and take advantage of specific hardware caracterísitcas that must support these data structures
Develop habits , attitudes and skills necessary to study independently or in teams, a specific subject , using available information sources (bibliography, Web , ...) and achieve the level of knowledge and understanding of the topic enough to explain it to others, writing a summary and preparing a complementary audiovisual material
Know and understand some basic principles for the design of computational experiments and learn basic techniques of data collection , validation and statistical analysis of the data collected and how to draw conclusions ; Recognizing the need , the usefulness and limitations of experimental studies in the design and implementation of algorithms and data structures

6. COURSE ORGANIZATION

	CONTENTS
1	1. Review of Basic Concepts Algorithmic Worst case analysis . Asymptotic notation. Divide and conquer. Analysis of recursive algorithms . Review of data structures : search trees , hashing, heaps . Graphs. Trail width and depth. Topological sorting .
2	Algorithmic Fundamental Concepts . Efficiency analysis . worst case and the average case. Landau notation . Divide and conquer. Analysis of recursive algorithms : Binary Search. Quicksort , MergeSort , QuickSetect . Analysis of Data Structures : Binary Search Trees . Balanced trees . Heaps . Tries . Analysis of greedy algorithms and associated structures : Dijkstra , Prim, Kruskal . Partitions. Hashing . Mounds of Fibonacci.
3	Probabilistic algorithms . Complexity classes BPP and ZPP . Probabilistic analysis . Examples: Rabin primality test , skip list , universal hashing .
4	Linear programming. Flow Networks. Zero-sum games . Evaluation Circuit . Simplex method : detailed algorithm and efficiency analysis .
5	Approximate algorithms and heuristics . Types approach , relative and absolute . Covering vertices . coverage metric vertices . Covering sets. Backpack. Local searches, Heuristics and metaheuristics

7. ASSESSMENT METHODS AND CRITERIA

Description	Type	Final Eval.	Reassessn	%
Laboratory practices	Others	No	Yes	20,00
Problems and exercices	Others	No	Yes	60,00
Group work and oral presentation	Work	No	No	20,00
TOTAL				100,00

Observations

.

Observations for part-time students

Students who complete the course part-time and not eligible for the regular assessment procedure may pass the course by attending a written exam (60 % of score) and laboratory examination of practices and exercises (40 % of the grade) . The minimum for each part average rating is 4 points out of 10 .

8. BIBLIOGRAPHY AND TEACHING MATERIALS

BASIC

KLEINBERG, J.; TARDOS, E. , Algorithm Design , Addison-Wesley , 2005 .

CORMEN, T.; LEISERSON, C.; RIVEST; R.; STEIN, C. , Introduction to Algorithms , The MIT Press , 2009 .

SEDGEWICK, R. , Algorithms in C++: Part 1-4 & Part 5 (3rd ed) , Addison-Wesley , 2002 .

DASGUPTA, S.; PAPADIMITRIOU, C.; VAZIRANI, U. , Algorithms , McGraw-Hill , 2008 .