Università degli Studi di L'Aquila - Dipartimento di Ingegneria e Scienze dell'Informazione e Matematica

Course catalogue

Page compiled on 21/07/2018

Programme of Integrated course "Distributed Systems and Web Algorithms"

This course is composed of 2 Modules: 1) Distributed Systems, 2) Web Algorithms

Programme of Module "Distributed Systems"

- Code: DT0168
- Type of course unit: Compulsory (Master Degree in Computer Science curriculum NEDAS), Elective (Master Degree in Computer Science curriculum SEAS), Elective (Master Degree in Computer Science curriculum UBIDIS)
- Level of course unit: Postgraduate Degrees
- Semester: 1

Number of ects credits: (Master Degree in Computer Science) 6 (workload 150 hours)

Teachers: Guido Proietti (Guido.Proietti@univaq.it)

1	**Course objectives**	The course provides the foundations for designing and analyzing (distributed) algorithms for reliable, faulty, concurrent, and adversarial distributed systems.
2	**Course content and learning outcomes (dublin descriptors)**	Topics of the module include: • Algorithms for COOPERATIVE Distributed Systems (DS) 1. Leader Election 2. Minimum Spanning Tree 3. Maximal Independent Set • Algorithms for UNRELIABLE DS: network monitoring, consensus problem • Algorithms for CONCURRENT DS: Mutual exclusion On successful completion of this module, the student should : • By the end of this module students will be able to: 1) understand the difference between a centralized and a distributed algorithm; 2) analyze the resources (space and time) needed by a distributed algorithm; 3) known efficient algorithms for basic computational distributed problems (leader election, consensus, etc.); 4) understand the difference between a canonical and a strategic distributed system. • The aim is to make the student capable of abstracting models and formal algorithmic problems from real distibuted computational problems, and designing efficient algorithmic solutions. • Through the presentation and the comparison of different solutions to a given probelm, students will be guided to learn and to identify independently their most efficient solution. • The course will encourage the development of the following skills of the student: capability of formally presenting and modelling concrete problems, focusing on their main features and discarding the inessential ones. • The course aims to develop in graduate students competencies and abilities necessary in their future studies, especially with respect to doctoral studies on algorithmic topics.
3	**Course prerequisites**	Knowledge of basic courses of discrete mathematcs and algorithms.
4	**Teaching methodsand language**	Mainly lectures, with only few exercises. **Language**: English **Reference textbooks** • P. Ferragina e F. Luccio, ***Crittografia***. Bollati Boringhieri. • H. Attiya e J. Welch, ***Distributed Computing***. Wiley.
5	**Assessment methods**	Mid-term written examination, followed by a final oral examination, which, for those who performed successfully in the mid-term examination, will be restricted to the second part of the course.

Programme of Module "Web Algorithms"

- Code: DT0167
- Type of course unit: Compulsory (Master Degree in Computer Science curriculum NEDAS), Elective (Master Degree in Computer Science curriculum SEAS), Compulsory (Master Degree in Computer Science curriculum

UBIDIS)
- Level of course unit: Postgraduate Degrees
- Semester: 1

Number of ects credits: (Master Degree in Computer Science) 6 (workload 150 hours)

Teachers: Guido Proietti (Guido.Proietti@univaq.it)

1	**Course objectives**	Knowledge of advanced algorithmic techniques; ability to individuate, formalize and solve optimization problems; concept of approximation; knowledge of the web search and sponsored web search strategies in search engines; ability to collaborate for the realization of applicative projects in group.
2	**Course content and learning outcomes (dublin descriptors)**	Topics of the module include: • Review of computational complexity and intractability. Optimization problems. Approximation algorithms. • Algorithmic techniques: greedy. local search, dynamic programming and linear programming. • Polynomial Time Approximation Schemes (PTAS) and Fully Polynomial Time Approximation Schemes (FPTAS). • Prestige and centrality indices in social networks. • Web search: Pagerank, Topical Pagerank, TrustRank, Hubs and Authorities. • Sponsored web search: matching markets and market clearing prices, auctions, VCG and GSP mechanisms. On successful completion of this module, the student should : • Acquire knowledge of advanced algorithmic techniques for NP-Hard optimization problems. In particular, the student will have mastery command of main algorithmic (approximation) techniques like greedy, local search, dynamic programming, linear programming: Polynomial Time Approximation Schemes (PTAS) and Fully Polynomial Time Approximation Schemes (FPTAS). Moreover the student will acquire knowledge on the basic centrality and prestige indices in social networks, on the main popularity indices for ranking pages in web search and finally of matching markets, auctions and the most important mechanisms adopted for the ranking and payment of sponsored search links. • Acquire the ability of abstracting models and formal algorithmic problems from real computational problems, understanding the degree of approximability and designing efficient algorithmic solutions. • Acquire autonomy in individuating, formalizing and understanding the degree of approximability of real computational problems and identify independently their most efficient solutions. • Being able to understand complex algorithmic solutions and to formal proving performances of their algorithmic solutions for complex computational problems. • Acquire the ability of understanding the ranking strategies adopted by search engines in web search and sponsored web search. • The course aims to develop in graduate students competencies and abilities necessary in their future studies and/or works, especially with respect to doctoral studies and in general to any research activity on algorithmic and web search topics.
3	**Course prerequisites**	KNOWLEDGE: fundamentals of programming, discrete mathematics, algorithms and data structures, computer architectures, reading and understanding of the English language SKILLS: ability to integrate classroom and homework study, ability to interact with the teacher during the class for originating discussion.
4	**Teaching methodsand language**	Lectures and exercises **Language**: English **Reference textbooks** • Vijay V. Vazirani, ***Approximation Algorithms***. Springer. 2001. • G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, M. Protasi, ***Complexity and Approximation***. Springer. 1999. • Jure Leskovec, Anand Rajaraman and Jeff Ullman, ***Mining of Massive Datasets***. Stanford University. 2011. *http://infolab.stanford.edu/~ullman/mmds/book.pdf* • Soumen Chakrabarti, ***Mining the Web – Discovering Knowledge from Hypertext***

		Data. Morgan Kaufmann. 2003. • David Easley and Jon Kleinberg, ***Networks, Crowds, and Markets***. Cambridge University Press. 2010. *https://www.cs.cornell.edu/home/kleinber/networks-book/*
5	**Assessment methods**	Written test followed by an oral exam. An optional mid-term written test will be also provided, which is meant to cover the first part of the course, in order to help the students to split the workload. If a student passes the mid-term written exam, she will take a final-term written exam concerned with the second part of the course content only. The mid-term written exam (lasting 2 hours) consists of exercises and open questions concerning the first part of the course content. The final-term written exam is split into two parts (each lasting one hour and half), each consisting of exercises and open questions, concerning the first and the second part of the course content, respectively. Students who passed the mid-term part will have to take only the second part. The final result of the written exam will be given by the average result of the two parts. The oral exam will occur within the same exam session of the written test, and it will typically cover the areas of the written answers that need clarification, plus a subject of one's choice. The oral exam (max 1 hour) will test the student's ability to engage in discussion of issues relevant to the topics discussed during the course. Criteria of evaluation will be the level of knowledge and the fluency in the technical language of algorithms and computational complexity.