



Programma del Corso integrato "Distributed Systems and Web Algorithms"

Il corso e' composto da 2 moduli: 1) Distributed Systems, 2) Web Algorithms

Programma del Modulo "Distributed Systems"

- Codice: DT0168
- Tipo di corso: Obbligatorio (Laurea Magistrale in Informatica percorso NEDAS), Opzionale (Laurea Magistrale in Informatica percorso SEAS), Opzionale (Laurea Magistrale in Informatica percorso UBIDIS)
- Livello del corso: Lauree Magistrali
- Semestre: 1

Numero di crediti ECTS: (Laurea Magistrale in Informatica) 6 (carico 150 ore)

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

1	<b>Obiettivi del corso</b>	Il corso fornisce gli elementi fondamentali di teoria e progettazione degli algoritmi in sistemi distribuiti, ovvero in sistemi in cui i soggetti computazionali coinvolti sono molteplici e possono perseguire o meno una strategia condivisa. Il corso spazierà quindi dalla presentazione dei classici algoritmi distribuiti per il problema dell'elezione del leader in sistemi cooperativi, fino ad arrivare agli algoritmi di crittografia, strumenti indispensabili per comprendere le vulnerabilità dei sistemi distribuiti aperti quali la rete Internet.
2	<b>Contenuti del corso e risultati formativi (descriptori di Dublino)</b>	<p>Gli argomenti trattati nel corso comprendono:</p> <ul style="list-style-type: none"> <li>• Algoritmi per sistemi distribuiti cooperativi: 1. Leader Election 2. Minimum Spanning Tree 3. Maximal Independent Set</li> <li>• Algoritmi per sistemi distribuiti con fallimenti: monitoraggio di una rete, problema del consenso</li> <li>• Algoritmi per sistemi distribuiti concorrenti: il problema della mutua esclusione</li> </ul> <p>Alla fine del corso, lo studente dovrebbe:</p> <ul style="list-style-type: none"> <li>• 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) know efficient algorithms for basic computational distributed problems (leader election, consensus, etc.); 4) understand the difference between a canonical and a strategic distributed system.</li> <li>• The aim is to make the student capable of abstracting models and formal algorithmic problems from real distributed computational problems, and designing efficient algorithmic solutions.</li> <li>• Through the presentation and the comparison of different solutions to a given problem, students will be guided to learn and to identify independently their most efficient solution.</li> <li>• 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.</li> <li>• 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.</li> </ul>
3	<b>Prerequisiti</b>	Conoscenza degli argomenti trattati nei corsi di matematica discreta e di algoritmi e strutture dati.
4	<b>Modalità e lingua di insegnamento</b>	<p>Prevalentemente lezioni teoriche, con alcune esercitazioni.</p> <p><b>Lingua:</b> Inglese</p> <p><b>Testi/Bibliografia</b></p> <ul style="list-style-type: none"> <li>• P. Ferragina e F. Luccio, <b>Crittografia</b>. Bollati Boringhieri.</li> <li>• H. Attiya e J. Welch, <b>Distributed Computing</b>. Wiley.</li> </ul>
5	<b>Metodi di accertamento</b>	Prova parziale scritta + Prova finale orale (eventualmente limitata alla seconda parte del corso nell'eventualità che sia stata superata con successo la prova parziale).

Programma del Modulo "Web Algorithms"

<ul style="list-style-type: none"> <li>• Codice: DT0167</li> <li>• Tipo di corso: Obbligatorio (Laurea Magistrale in Informatica percorso NEDAS), Opzionale (Laurea Magistrale in Informatica percorso SEAS), Obbligatorio (Laurea Magistrale in Informatica percorso UBIDIS)</li> <li>• Livello del corso: Lauree Magistrali</li> <li>• Semestre: 1</li> </ul>	
Numero di crediti ECTS: (Laurea Magistrale in Informatica) 6 (carico 150 ore)	
Docenti: Cosimo Vinci (cosimo.vinci@univaq.it)	
1	<p><b>Obiettivi del corso</b></p> <p>Conoscenza di tecniche algoritmiche avanzate, capacità di individuazione, formalizzazione e risoluzione di problemi di ottimizzazione, concetto di approssimazione, conoscenza delle strategie di web search e sponsored web search nei motori di ricerca, capacità di collaborazione per la realizzazione progetti applicativi di gruppo</p>
2	<p><b>Contenuti del corso e risultati formativi (descrittori di Dublino)</b></p> <p>Gli argomenti trattati nel corso comprendono:</p> <ul style="list-style-type: none"> <li>• Richiami di complessità ed intrattabilità. Problemi di ottimizzazione. Algoritmi di approssimazione.</li> <li>• Tecniche algoritmiche: greedy, ricerca locale, programmazione dinamica e programmazione lineare.</li> <li>• Schemi di approssimazione polinomiali (PTAS) e pienamente polinomiali (FPTAS).</li> <li>• Indici di centralità e prestigio nelle reti sociali</li> <li>• Web search: Pagerank, Topical Pagerank, TrustRank, Hubs e Authorities</li> <li>• Sponsored web search: matching markets e market clearing prices, aste, meccanismi VCG e GSP.</li> </ul> <p>Alla fine del corso, lo studente dovrebbe:</p> <ul style="list-style-type: none"> <li>• 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.</li> <li>• Acquire the ability of abstracting models and formal algorithmic problems from real computational problems, understanding the degree of approximability and designing efficient algorithmic solutions.</li> <li>• Acquire autonomy in individuating, formalizing and understanding the degree of approximability of real computational problems and identify independently their most efficient solutions.</li> <li>• Being able to understand complex algorithmic solutions and to formal proving performances of their algorithmic solutions for complex computational problems.</li> <li>• Acquire the ability of understanding the ranking strategies adopted by search engines in web search and sponsored web search.</li> <li>• 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.</li> </ul>
3	<p><b>Prerequisiti</b></p> <p>CONOSCENZE : fondamenti di programmazione, matematica discreta, algoritmi e strutture dati, architetture degli elaboratori, lettura e comprensione in lingua inglese CAPACITA' : capacità di integrazione dello studio in aula con lo studio personale, capacità di interazione con il docente in aula in modo da originare momenti comuni di confronto</p>
4	<p><b>Modalità e lingua</b></p> <p>Lezioni frontali e esercizi</p>

	<b>di insegnamento</b>	<b>Lingua:</b> Inglese <b>Testi/Bibliografia</b> <ul style="list-style-type: none"><li>• Vijay V. Vazirani, <b>Approximation Algorithms</b>. Springer. 2001.</li><li>• G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, M. Protasi, <b>Complexity and Approximation</b>. Springer. 1999.</li><li>• Jure Leskovec, Anand Rajaraman and Jeff Ullman, <b>Mining of Massive Datasets</b>. Stanford University. 2011. <a href="http://infolab.stanford.edu/~ullman/mmds/book.pdf">http://infolab.stanford.edu/~ullman/mmds/book.pdf</a></li><li>• Soumen Chakrabarti, <b>Mining the Web – Discovering Knowledge from Hypertext Data</b>. Morgan Kaufmann. 2003.</li><li>• David Easley and Jon Kleinberg, <b>Networks, Crowds, and Markets</b>. Cambridge University Press. 2010. <a href="https://www.cs.cornell.edu/home/kleinber/networks-book/">https://www.cs.cornell.edu/home/kleinber/networks-book/</a></li></ul>
5	<b>Metodi di accertamento</b>	Prova scritta ed un eventuale esame orale. L'esame scritto consiste in due distinte prove scritte (una sulla parte di algoritmi di approssimazione, e l'altra su web search e sponsored web search), della durata di 1.45 minuti ognuna. A discrezione dello studente, le due prove scritte possono essere sostenute in appelli differenti. Ogni prova deve essere superata con un voto di 18/30 e il voto finale sarà dato dalla media delle due parti. Un eventuale esame orale consiste in una discussione dettagliata della prova scritta, più qualche eventuale domanda relativa alla parte teorica del corso.