



UNIVERSITÀ
DEGLI STUDI
DI PADOVA



Bollettino Notiziario - A.A. 2013/2014

LAUREA MAGISTRALE IN INFORMATICA (ORD. 2009)

Curriculum: Corsi comuni

ALGORITMI DI APPROSSIMAZIONE

Titolare: Prof. LIVIO COLUSSI

Periodo: Il anno, 1 trimestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 40A+8E; 6,00

Sede dell'insegnamento: Informazioni in lingua non trovate

Aule: Informazioni in lingua non trovate

Prerequisiti:

?????Conoscenze di base di algoritmi e strutture dati, delle principali tecniche algoritmiche e dell'analisi della complessità degli algoritmi. L'insegnamento non prevede propedeuticità

Conoscenze e abilità da acquisire:

Per molti problemi computazionali di interesse pratico si sa che non esistono algoritmi efficienti per la loro risoluzione. Tali problemi si possono quindi risolvere soltanto per istanze molto piccole ma non nei casi pratici di interesse. In questo caso si può talvolta ricorrere ad algoritmi di approssimazione i quali calcolano soltanto una "approssimazione" della soluzione del problema ma fanno ciò in modo molto più efficiente e quindi risultano utilizzabili effettivamente nei casi pratici. In questo corso si studieranno le tecniche per ottenere degli algoritmi di approssimazione.

Attività di apprendimento previste e metodologie di insegnamento:

Lezioni frontali.

Contenuti:

Prima parte: Algoritmi on-line. - Analisi competitiva per gli algoritmi on-line. - Paginazione: Competitività di LRU. Algoritmo off-line ottimo. Limite inferiore per la competitività degli algoritmi on-line. Algoritmi on-line randomizzati. Analisi di MARKING. Limite inferiore per algoritmi on-line randomizzati. Tipi di avversari e competitività contro i vari tipi di avversari. Analisi di RANDOM. - K server: Riassunto dei risultati noti e algoritmi on-line ingenui. ATTIVI, un algoritmo k-competitivo sugli alberi. RWALK, un algoritmo k-competitivo sugli spazi metrici resistivi. L'algoritmo della funzione lavoro (2k-1)-competitivo su spazi metrici generali (senza dimostrazione della competitività). Seconda parte: Algoritmi di approssimazione. - Risultati negativi: Caratterizzazioni delle classi P ed NP in termini di verificatori. Teorema di Cook: 3SAT è NP-completo (senza dimostrazione). Riduzione di 3SAT a 3COLOR. Verificatori probabilistici. Il teorema PCP di Arora (senza dimostrazione). Caratterizzazione di Arora della classe NP in termini di verificatori probabilistici. Risultati di non approssimabilità derivati dal teorema di Arora. Caratterizzazione di Fagin della classe NP e problemi MAX-SNP-completi. - Progetto di algoritmi approssimati: Algoritmo di Cristofides per il problema TSP euclideo. La tecnica del rilassamento. Rilassamento di tipo LP. Il problema del minimo ricoprimento di vertici. Algoritmi di Hochbaum e di Bar-Yehuda ed Even. Il metodo primale-duale. Il problema del matching perfetto di costo minimo. Algoritmo di Goemans e Williamson. Uso di rilassamenti non lineari. Il problema del taglio massimo. L'algoritmo 0.878. - Gli schemi di approssimazione e le classi PAS, FPAS, PAAS ed FPAAS. I problemi dell'impacchettamento e della schedulazione e il problema di decisione associato. Comportamento diverso dei due problemi rispetto all'approssimabilità. Un PAS per il problema della schedulazione.

Modalità di esame:

Esame orale.

Criteri di valutazione:

La prova orale accerterà la conoscenza degli algoritmi e delle tecniche algoritmiche spiegate a lezione.

Testi di riferimento:

David P. Williamson and David B. Shmoys, The Design of Approximation Algorithms. : Cambridge University Press, 2010 Vijay V. Vazirani, Approximation algorithms. : Springer, 2001

Eventuali indicazioni sui materiali di studio:

Dispense del docente.

ALTRE ATTIVITA' UTILI PER IL LAVORO (O TIROCINIO)

Titolare: da definire

Periodo: Il anno, 3 trimestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: ; 2,00

AMMINISTRAZIONE DI SISTEMA

Titolare: Dott. FRANCESCO CLABOT

Periodo: I anno, 3 trimestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 40A+8E; 6,00

Prerequisiti:

Il corso non prevede particolari prerequisiti.

Conoscenze e abilità da acquisire:

Il corso si propone di presentare agli studenti l'organizzazione di un dipartimento ICT di una grande azienda. In particolare verranno trattate tematiche legate alle metodologie consolidate per l'impostazione dei processi ICT (ITIL), le motivazioni che sono alla base delle scelte dei prodotti e tecnologie adottate (ROI, SLA, etc.), esempi concreti di architetture informatiche basilari oltre a vari case studies.

Attività di apprendimento previste e metodologie di insegnamento:

Lezioni frontali e laboratorio

Contenuti:

- La gestione dei servizi informatici (ITIL): i processi coinvolti nelle due aree della Gestione dei Servizi (Service Support e Service Delivery), la loro applicazione al ciclo operativo completo dei servizi, gli obiettivi fondamentali e perchè questi sono stati standardizzati, breve dissertazione su ognuno dei 10 servizi coinvolti, esempi pratici. - Modelli di servizio: considerazioni su ROI e SLA, approccio ed aspetti pratici. - Il dipartimento IT: struttura ed organizzazione. Organigramma generale e breve dissertazione sui vari settori. Analisi accurata del "Service Desk" (come evoluzione dell'Help Desk). - L'infrastruttura informatica: in verticale dal network ai servizi richiamando sempre i concetti esposti nella prima parte del corso. Esempi pratici (no laboratorio) e case study per mettere alla prova le capacità deduttive degli studenti

Modalità di esame:

L'esame finale consisterà in un test scritto composto da 40 domande a scelta multipla.

Criteri di valutazione:

Le conoscenze dello studente vengono valutate mediante un test a risposta multipla. La votazione finale prenderà in considerazione anche la qualità dell'attività di laboratorio condotta.

Testi di riferimento:

Jan Van Bon, Foundations of IT Service Management-based on ITIL... : Van Haren Publishing,

Eventuali indicazioni sui materiali di studio:

Sul sito web del corso (link da <http://www.netadm.it>) sono presenti molti documenti scaricabili in formato digitale: case study, articoli divulgativi etc.

ANALISI NUMERICA

Titolare: Prof. MARCO VIANELLO

Mutuato da: Laurea magistrale in Astronomia (Ord. 2010)

Periodo: I anno, 1 trimestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 40A+16E; 6,00

Sede dell'insegnamento: Torre Archimede

Aule: Torre Archimede

Conoscenze e abilità da acquisire:

Apprendere le basi del calcolo numerico in vista delle applicazioni in campo scientifico e tecnologico, con particolare attenzione ai concetti di errore, discretizzazione, approssimazione, convergenza, stabilità, costo computazionale

Attività di apprendimento previste e metodologie di insegnamento:

Sistema-floating point e propagazione degli errori: errore di troncamento e di arrotondamento, rappresentazione floating-point dei reali, precisione di macchina, operazioni aritmetiche con numeri approssimati, condizionamento di funzioni, propagazione degli errori in algoritmi iterativi per esempi, il concetto di stabilità. Soluzione numerica di equazioni non lineari: metodo di bisezione, stima dell'errore col residuo pesato; metodo di Newton,

convergenza globale, velocità di convergenza, convergenza locale, stima dell'errore, altri metodi di linearizzazione; iterazioni di punto fisso Interpolazione e approssimazione di funzioni e dati: interpolazione polinomiale, interpolazione di Lagrange, errore di interpolazione, il problema della convergenza (controesempio di Runge), interpolazione di Chebyshev, stabilità dell'interpolazione; interpolazione polinomiale a tratti, interpolazione spline; approssimazione polinomiale ai minimi quadrati Integrazione e derivazione numerica: formule algebriche e composte, convergenza e stabilità, esempi; instabilità dell'operazione di derivazione, calcolo di derivate tramite formule alle differenze; il concetto di estrapolazione Elementi di algebra lineare numerica: norme di vettori e matrici, condizionamento di matrici e sistemi; metodi diretti: metodo di eliminazione gaussiana e fattorizzazione LU, calcolo del determinante, calcolo della matrice inversa, fattorizzazione QR, soluzione ai minimi quadrati di sistemi sovradeterminati; metodi iterativi: i metodi di Jacobi e Gauss-Seidel, struttura generale delle iterazioni stazionarie, preconditionamento; metodo delle potenze per il calcolo di autovalori estremali Introduzione ai metodi alle differenze finite per equazioni differenziali: i metodi di Eulero esplicito ed implicito, il metodo trapezoidale, convergenza e stabilità, sistemi stiff; equazione di Poisson 1d e 2d; metodo delle linee per l'equazione del calore Laboratorio: implementazione e applicazione di codici numerici in Matlab

Contenuti:

Sistema floating-point e propagazione degli errori Soluzione numerica di equazioni non lineari Interpolazione e approssimazione di dati e funzioni Integrazione e derivazione numerica Elementi di algebra lineare numerica Introduzione ai metodi alle differenze finite per equazioni differenziali

Modalità di esame:

Lezioni in aula e laboratorio

Criteri di valutazione:

Prova orale

Testi di riferimento:

A. Quarteroni, F. Saleri, Introduzione al calcolo scientifico. : Springer, A. Quarteroni, F. Saleri, Scientific computing with Matlab and Octave. : Springer, G. Rodriguez, Algoritmi numerici. : Pitagora,

ANALISI STATICA E VERIFICA DEL SOFTWARE

Titolare: Prof. FRANCESCO RANZATO

Periodo: I anno, 1 trimestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 64A; 8,00

Sede dell'insegnamento: Torre Archimede, Padova

Prerequisiti:

Conoscenze di base dei linguaggi di programmazione. L'insegnamento non prevede propedeuticità.

Conoscenze e abilità da acquisire:

Il corso mira ad introdurre metodi e strumenti per la specifica del comportamento, l'analisi statica e la verifica automatica dei programmi e, più in generale, dei sistemi software. In particolare, il corso fornisce una introduzione alla semantica formale dei linguaggi di programmazione ed ai metodi formali per la loro analisi statica e verifica.

Attività di apprendimento previste e metodologie di insegnamento:

L'insegnamento prevede lezioni frontali e la risoluzione in modo indipendente a casa di vari esercizi.

Contenuti:

- Semantica operativa di programmi: Modellazione del comportamento operativo dei programmi su una macchina di esecuzione mediante sistemi di regole di derivazione. - Semantica denotazionale di programmi: Modellazione del comportamento input/output dei programmi mediante la teoria degli ordini parziali e dei punti fissi. - Analisi statica di programmi mediante interpretazione astratta: L'interpretazione astratta è una nota tecnica basata su una approssimazione della semantica denotazionale dei programmi che permette di specificare le proprietà dei programmi deducibili mediante analisi statica e di provarne la correttezza. - Analisi statica dataflow di programmi: tecnica per dedurre staticamente informazioni sull'insieme dei possibili valori delle variabili nei vari punti del programma. Un grafo di flusso del controllo è utilizzato per determinare le parti di un programma a cui un particolare valore assegnato ad una variabile potrebbe propagarsi. Le informazioni raccolte sono spesso utilizzate dai compilatori per ottimizzare un programma. - Verifica di sistemi software mediante model checking: Il model checking è una tecnica per la verifica automatica di proprietà di correttezza di un sistema software, dove la correttezza è specificata mediante logiche temporali. Gli inventori del model checking sono stati premiati con il prestigioso Turing Award (noto come il "Premio Nobel" dell'informatica) nel 2007.

Modalità di esame:

Esame orale, tipicamente suddiviso in tre parti distinte.

Criteri di valutazione:

L'esame orale verte su vari esercizi che lo studente deve svolgere in modo indipendente a casa.

Testi di riferimento:

H. Riis Nielson, F. Nielson, Semantics with Applications: A Formal Introduction. : Wiley, 1992

Eventuali indicazioni sui materiali di studio:

Le slide utilizzate a lezione verranno distribuite.

APPRENDIMENTO AUTOMATICO

Titolare: Prof. ALESSANDRO SPERDUTI

Periodo: I anno, 3 trimestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 40A+8L; 6,00

Prerequisiti:

È opportuno avere familiarità con le conoscenze matematiche relative al Calcolo delle Probabilità e all'Analisi di funzioni multivariate. Inoltre è consigliabile avere conoscenze di base relative alla Programmazione e all'Intelligenza Artificiale. L'insegnamento non prevede propedeuticità.

Conoscenze e abilità da acquisire:

In questo insegnamento si presentano alcuni dei concetti fondamentali che caratterizzano l'Apprendimento Automatico, cioè quella classe di tecniche ed algoritmi che a partire da dati empirici permettono di acquisire nuova conoscenza, oppure di correggere e/o raffinare conoscenza già disponibile. Tali tecniche sono particolarmente utili per problemi per cui è impossibile o molto difficile pervenire ad una formalizzazione utilizzabile per la definizione di una soluzione algoritmica ad hoc. Esempi di tali problemi sono compiti percettivi, come il riconoscimento visivo di cifre manoscritte, e problemi in cui i dati sono corrotti dal rumore o sono incompleti. L'insegnamento tratta principalmente metodi numerici. Sono previste esercitazioni in laboratorio informatico che consentono allo studente di sperimentare le conoscenze acquisite mediante l'applicazione a piccoli esempi pratici.

Attività di apprendimento previste e metodologie di insegnamento:

L'insegnamento prevede lezioni frontali ed esercitazioni in laboratorio informatico. Le esercitazioni in laboratorio informatico consistono nella sperimentazione da parte degli studenti delle tecniche viste a lezione sotto vari scenari operativi. In questo modo gli studenti possono verificare sperimentalmente i concetti appresi e acquisire sia capacità di applicazione dei concetti appresi che di giudizio critico.

Contenuti:

La struttura e le tematiche dell'insegnamento saranno le seguenti: - Introduzione: Quando Applicare le Tecniche Proprie dell'Apprendimento Automatico; Paradigmi di Apprendimento Automatico; Gli ingredienti Fondamentali dell'Apprendimento Automatico. - Apprendimento di Concetti: Complessità dello Spazio delle Ipotesi; Misure di Complessità; Esempi di Algoritmi di Apprendimento Supervisionato; - Alberi di Decisione: Apprendimento di Alberi di Decisione; Trattamento di Dati Numerici, di Dati Mancanti, di Costi; Tecniche di Pruning e Derivazione di Regole di Decisione. - Apprendimento Probabilistico: Apprendimento Bayesiano; Esempi di Applicazione al Paradigma Supervisionato e al Paradigma Non-Supervisionato (clustering); Classificatore Ottimo di Bayes; EM. - Reti Neurali e Support Vector Machines: Cenni di Reti Neurali; Margine di Classificazione; Support Vector Machines per Classificazione e Regressione; Funzioni Kernel. - Aspetti Applicativi: Pipeline di Classificazione; Rappresentazione e Selezione di Variabili Categoriche; Model Selection, Holdout, CrossValidation, LeaveOneOut CV; Criteri Esterni e Interni per Valutare un Sistema di Clustering; Sistemi di Raccomandazione: Tipologie, Approcci, Misure di Valutazione.

Modalità di esame:

Lo studente deve superare un esame scritto e, se ritenuto necessario dal docente, un esame orale.

Criteri di valutazione:

Il testo dell'esame scritto contiene alcune domande che consentono di valutare il livello di apprendimento delle nozioni impartite durante l'insegnamento e la capacità dello studente nell'analizzarle criticamente. Sono poi presenti domande in cui si richiede allo studente di mostrare di aver compreso gli aspetti applicativi trattati all'interno delle attività svolte in laboratorio informatico. Tali domande hanno lo scopo di valutare se lo studente ha sviluppato la capacità di applicare le nozioni apprese durante l'insegnamento. Nel caso in cui la valutazione dello scritto non risulti soddisfacente per lo studente, il docente può integrare l'esame scritto con un esame orale per meglio verificare la preparazione dello studente.

Testi di riferimento:

Tom Mitchell, Machine Learning. : McGraw Hill, 1998 Ethem Alpaydin, Introduction to Machine Learning. : Cambridge University Press, 2010

Eventuali indicazioni sui materiali di studio:

Vengono rese disponibili, come riferimento, i lucidi utilizzati a lezione.

BIOINFORMATICA

Titolare: Prof. GIORGIO VALLE

Periodo: I anno, 2 trimestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 40A; 6,00

Prerequisiti:

Non ci sono prerequisiti particolari, se non quanto ci si aspetta da uno studente magistrale di informatica. Una conoscenza di base della genetica e della biologia molecolare saranno comunque utili per meglio inquadrare le motivazioni biologiche che stanno alla base della bioinformatica. Il corso è in lingua inglese, quindi è necessario avere una buona conoscenza dell'inglese scritto e parlato.

Conoscenze e abilità da acquisire:

Il Corso è suddiviso in tre parti principali: la prima parte mette in relazione Biologia e Informazione; la seconda parte descrive i principali algoritmi utilizzati in bioinformatica per allineare sequenze biologiche e assemblare genomi; la terza parte tratta di problemi di bioinformatica relativi alla genomica funzionale. Inoltre il corso è accompagnato da esercitazioni pratiche in cui gli studenti applicheranno metodi bioinformatici per analizzare dati genomici. In considerazione della complessità della materia e in accordo con i descrittori di Dublino, particolare attenzione sarà dedicata affinché gli studenti acquisiscano la capacità di integrare le conoscenze e gestire la complessità dei problemi trattati, nonché di formulare giudizi sulla base di informazioni limitate e spesso frammentarie.

Attività di apprendimento previste e metodologie di insegnamento:

Il corso sarà tenuto con lezioni frontali e con esercitazioni pratiche. Sarà stimolata la discussione in classe.

Contenuti:

Questo è un corso di 6 crediti: cinque di lezioni ed uno di attività pratiche che consistono nell'implementazione di algoritmi oppure in un'approfondita indagine della letteratura, su argomenti assegnati. Le lezioni sono organizzate in tre parti. La prima parte è un'approfondita introduzione alla Biologia, presentata come una disciplina scientifica centrata sull'informazione. I meccanismi che facilitano la trasmissione e l'evoluzione dell'informazione biologica saranno presi come spunto per introdurre alcuni problemi della biologia che richiedono approcci computazionali e strumenti bioinformatici. La seconda parte del corso descrive i principali algoritmi utilizzati per allineare sequenze biologiche, inclusi quelli sviluppati per il sequenziamento di DNA di ultima generazione. Sono inoltre descritti gli algoritmi utilizzati per l'assemblaggio "de novo" di genomi. Infine, la terza parte del corso copre alcuni aspetti della bioinformatica relativi alla genomica funzionale, come l'analisi del trascrittoma, la predizione e annotazione genica, la ricerca di pattern e motivi per la predizione delle strutture proteiche. Inoltre viene discusso il ruolo della bioinformatica nell'analisi di genomi individuali e nella medicina personalizzata.

Modalità di esame:

L'esame sarà orale, ma un continuo monitoraggio sarà attuato durante l'intera durata del corso per verificare la comprensione degli studenti.

Criteri di valutazione:

Nell'esame finale gli studenti dovranno dimostrare una comprensione sistematica del settore e dovranno sapersi destreggiare con i metodi della ricerca associati ad esso. Inoltre gli studenti dovrebbero essere capaci di analisi critica, di valutare e sintetizzare idee nuove e complesse, integrando gli argomenti di questo corso con altre conoscenze.

Testi di riferimento:

CONTENUTO NON PRESENTE

Eventuali indicazioni sui materiali di studio:

Non sono previsti libri ufficiali di testo e gli studenti saranno stimolati a trovare le informazioni su fonti multiple. Materiale didattico con gli approfondimenti di quanto spiegato a lezione sarà disponibile sul sito web del docente: <http://didattica.cribi.unipd.it/genomica/bioinforinfo/>.

COMPUTABILITA' E ALGORITMI

Titolare: Prof. PAOLO BALDAN

Periodo: I anno, 2 trimestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 64A+16E; 10,00

Prerequisiti:

Il corso richiede familiarità con alcuni concetti matematici di base, quali relazioni, funzioni, insiemi, cardinalità, ordini parziali, principi di induzione. Non ci sono corsi propedeutici.

Conoscenze e abilità da acquisire:

Obiettivo del corso è quello di avvicinare lo studente ai temi classici della teoria della calcolabilità e di completare e approfondire le conoscenze algoritmiche fondamentali acquisite nella laurea di primo livello. Per la prima parte, partendo dall'esame matematico del concetto di procedimento effettivo, si studiano i limiti che tale nozione impone sulla classe delle funzioni effettivamente calcolabili da un algoritmo, con lo sviluppo di una teoria dell'indecidibilità e della ricorsione. Per la seconda parte si approfondiscono alcune tecniche algoritmiche per l'elaborazione di strutture fondamentali quali grafi, stringhe e oggetti geometrici, si studiano algoritmi multithread e randomizzati. A livello più generale, il corso mira ad implementare le capacità di formalizzazione, ragionamento e problem solving dello studente.

Attività di apprendimento previste e metodologie di insegnamento:

Il corso prevede lezioni frontali ed esercizi.

Contenuti:

Il corso si articola in due parti, la prima focalizzata sulla teoria della computabilità, e la seconda che approfondisce tematiche di natura prettamente algoritmica. Per quanto riguarda la teoria della computabilità saranno sviluppati i seguenti temi: - Algoritmi ed il concetto di procedimento effettivo. Macchine a registri (URM). Funzioni parziali ricorsive. Equivalenze tra modelli di calcolo. Universalità dei modelli di calcolo. Tesi di Church. - Enumerazione delle funzioni calcolabili. Esistenza di funzioni non calcolabili: il metodo della diagonalizzazione. Il teorema del parametro. Programmi universali. - Problemi decidibili, indecidibili e semidecidibili. Indecidibilità del problema della fermata. Metodo di riduzione. Esempi di altri problemi indecidibili. - Insiemi ricorsivi e ricorsivamente enumerabili. Teoremi di Rice e di Rice-Shapiro. - Funzionali. Definizioni ricorsive. Ordinamenti parziali, funzioni monotone e punti fissi. Funzionali ricorsivi. Il teorema di Myhill-Sheperdson. Primo teorema di ricorsione. Secondo teorema di ricorsione. L'approfondimento delle tecniche algoritmiche si concentrerà su: - Algoritmi su grafi. Visita in ampiezza e visita in profondità. Ordinamento topologico. Componenti fortemente connesse. - Algoritmi su stringhe. Algoritmi basati su confronti (Knuth, Morris e Pratt, di Boyer, Moore e Yao, Corasich). Algoritmi seminumerici (ShiftAnd e Fingerprint di Rabin, Karp). Alberi dei suffissi e algoritmo di Ukkonen per la loro costruzione in tempo lineare. - Algoritmi Multithread. - Algoritmi di Geometria Computazionale. Rappresentazione degli oggetti geometrici e algoritmi di base. Test di non intersezione tra segmenti. Involucro convesso: algoritmi di Graham e di Jarvis. Localizzazione di un punto in un piano suddiviso in regioni poligonali. - Algoritmi randomizzati. Algoritmo di rendering. Algoritmo di routing.

Modalità di esame:

L'esame si articola in una prova scritta, principalmente focalizzata sullo svolgimento di esercizi di teoria della computabilità, e in una discussione orale sulle tecniche algoritmiche.

Criteri di valutazione:

La prova scritta contiene esercizi atti a verificare la capacità dello studente di utilizzare nozioni e tecniche dimostrative apprese durante il corso, per la soluzione di problemi nuovi. La prova orale verifica la conoscenza ed il livello di approfondimento dei temi trattati a lezione, con la descrizione di nozioni e la riproduzione di dimostrazioni note.

Testi di riferimento:

Nigel Cutland, Computability. An Introduction to Recursive Function Theory. : Cambridge University Press, 1980 T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduzione agli Algoritmi e Strutture Dati (3a edizione). : McGraw-Hill Italia, 2010

Eventuali indicazioni sui materiali di studio:

CRITTOGRAFIA

Titolare: Prof. ALESSANDRO LANGUASCO

Periodo: I anno, 1 trimestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 40A+8E; 6,00

Prerequisiti:

Gli argomenti dei corsi di Algebra, Analisi I e Algoritmi (in particolare per i calcoli della loro complessita' computazionale).

Conoscenze e abilità da acquisire:

Lo scopo del corso e' quello di offrire una panoramica delle basi teoriche necessarie per permettere uno studio critico dei protocolli crittografici usati oggi in molte applicazioni (autenticazione, commercio digitale). Nella prima parte verranno esposti gli strumenti matematici di base (essenzialmente dalla teoria elementare ed analitica dei numeri) necessari per comprendere il funzionamento dei moderni metodi a chiave pubblica. Nella seconda parte vedremo come applicare queste conoscenze per studiare in modo critico alcuni protocolli crittografici.

Attività di apprendimento previste e metodologie di insegnamento:

Lezione frontale.

Contenuti:

First Part: Basic theoretical facts: Modular arithmetic. Prime numbers. Little Fermat theorem. Chinese remainder theorem. Finite fields: order of an element and primitive roots. Pseudoprimality tests. Agrawal-Kayal-Saxena's test. RSA method: first description, attacks. Rabin's method and its connection with the integer factorization. Discrete logarithm methods. How to compute the discrete log in a finite field. Elementary factorization methods. Some remarks on Pomerance's quadratic sieve. Second Part: Protocols and algorithms. Fundamental crypto algorithms. Symmetric methods (historical ones, DES, AES) . Asymmetric methods. Attacks. Digital signature. Pseudorandom generators (remarks). Key exchange, Key exchange in three steps, secret splitting, secret sharing, secret broadcasting, timestamping. Signatures with RSA and discrete log.

Modalità di esame:

Scritto

Criteri di valutazione:

Durante la prova scritta lo studente dovrà rispondere ad alcune domande relative al programma svolto dimostrando di aver compreso gli argomenti del corso. Il massimo dei voti (30/30) verrà assegnato in presenza di un compito privo di errori. Il docente si riserva di fare alcune domande orali nel caso in cui sia necessario investigare ulteriormente la preparazione del candidato.

Testi di riferimento:

Languasco - Zaccagnini, Introduzione alla Crittografia. Milano: Hoepli, 2004

Eventuali indicazioni sui materiali di studio:

Utilizzeremo i seguenti testi: 1) A.Languasco, A.Zaccagnini - Introduzione alla Crittografia - Hoepli Editore, 2004. (italian). 2) N.Koblitz - A Course in Number Theory and Cryptography, Springer, 1994. 3) R.Crandall, C.Pomerance, - Prime numbers: A computational perspective - Springer, 2005. 4) B. Schneier - Applied Cryptography - Wiley, 1994

DATA MINING

Titolare: Prof. BRUNO SCARPA

Periodo: I anno, 2 trimestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 34A+16L; 6,00

Prerequisiti:

Conoscenze di Informatica di base, Basi di Dati

Conoscenze e abilità da acquisire:

Il corso intende fornire una panoramica sui concetti e sulle metodologie e strumenti avanzati di analisi di grandi quantità di dati, spesso usate come supporto al processo di decisione aziendale.

Attività di apprendimento previste e metodologie di insegnamento:

Lezioni frontali, laboratori con analisi di dati reali

Contenuti:

- L'analisi dei dati come strumento di supporto per le decisioni e la Business Intelligence. Motivazioni e contesto per il data mining. - I modelli statistici: modelli lineari e GLM, la stima e l'adattamento ai dati - Nozioni generali per il data mining: contrasto tra aderenza ai dati e complessità del modello ovvero contrasto tra distorsione e varianza, tecniche generali per la selezione del modello (AIC, BIC, convalida incrociata, oltre ai test statistici classici), suddivisione dei dati in un insieme di lavoro e uno di verifica. - Metodi di regressione: regressione non parametrica, modelli additivi, alberi, mars, projection pursuit, reti neurali (richiami). - Metodi di classificazione: mediante la regressione lineare, regressione logistica e multilogit, modelli additivi, alberi, polymars, reti neurali, combinazione di classificatori (bagging, boosting, foreste casuali). - Metodi di analisi interna: nozioni sui metodi di raggruppamento, analisi delle associazioni tra variabili e algoritmo Apriori. Reti sociali (cenni).

Modalità di esame:

Scritta/Pratica (con eventuale progetto)

Criteri di valutazione:

Le prove d'esame misureranno quanto ciascuno studente (a) saprà e quanto (b) saprà applicare degli strumenti proposti durante il corso.

Testi di riferimento:

Azzalini A., Scarpa B., Analisi dei dati e data mining. : Springer, 2004 Azzalini A., Scarpa B., Data analysis and data mining. : Oxford University Press, 2012

Eventuali indicazioni sui materiali di studio:

Libro di testo e materiale didattico fornito dal docente.

FONDAMENTI LOGICI DEI LINGUAGGI FUNZIONALI

Titolare: Prof. SILVIO VALENTINI

Periodo: I anno, 1 trimestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 40A+8E; 6,00

Prerequisiti:

Conoscenze di base di logica matematica e del linguaggio insiemistico

Conoscenze e abilità da acquisire:

Lo scopo di questo corso è quello di fornire una introduzione teorica ai linguaggi di programmazione funzionali tipati e non tipati.

Attività di apprendimento previste e metodologie di insegnamento:

Lezioni frontali in aula

Contenuti:

Dopo aver richiamato la nozione di funzione calcolabile si introdurrà il lambda calcolo puro e si dimostrerà che esso è uno strumento universale di calcolo. Si analizzerà quindi il lambda calcolo tipato semplice ed i suoi legami con il frammento implicativo del calcolo proposizionale intuizionista. Si introdurranno poi il calcolo con tipi dipendenti, che rappresenta il contenuto computazionale della logica del primo ordine, per continuare con calcoli con tipi di secondo ordine, potenti quanto l'aritmetica di Heyting al secondo ordine, e finire quindi con calcoli estremamente potenti che considerano insieme entrambi i sistemi di tipi ed eventualmente anche i tipi induttivamente generati, i tipi ricorsivi ed i tipi intersezione. Per tutti tali lambda calcoli si intendono dimostrare i principali teoremi matematici, vale a dire il teorema di normalizzazione e di confluenza, e fornire esempi di applicazione in informatica teorica.

Modalità di esame:

L' accertamento di profitto avverrà con una prova orale dopo il completamento di esercitazioni personali da parte dello studente.

Criteri di valutazione:

L'esame intende valutare le conoscenze acquisite dallo studente sui temi del corso e le sue capacità di svolgere del lavoro autonomo su di essi.

Testi di riferimento:

H.Barendreght, The Lambda Calculus, its Syntax and Semantics. : North-Holland, J.Y.Girard, Y.Lafont, P.Taylor, Proofs and Types. : Cambridge University Press, H.Barendreght,, Lambda Calculi with Types. : Oxford University Press,

Eventuali indicazioni sui materiali di studio:

Appunti forniti dal docente

GESTIONE DI IMPRESE INFORMATICHE

Titolare: Prof. AMIR BALDISSERA

Periodo: Il anno, 2 trimestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 32A+16E; 6,00

Prerequisiti:

Il corso non ha prerequisiti.

Conoscenze e abilità da acquisire:

Fornire allo studente le basi teoriche e pratiche sull'ideazione e la gestione di un progetto di business. Particolare attenzione verrà riservata ai progetti legati al mondo dell'informatica ed alle nuove tecnologie. Si affronterà il tema delle startup digitali e delle loro dinamiche. Al termine del corso lo studente potrà avere tutti gli strumenti per ideare, valutare e avviare un progetto di business efficace.

Attività di apprendimento previste e metodologie di insegnamento:

Lezioni frontali forniranno le basi teoriche e numerosi esempi pratici serviranno per mostrare come le diverse aziende operano sul mercato. La realizzazione di un progetto proporrà la pianificazione del lancio di una startup, dalla creazione del team, all'ideazione alla creazione del modello di business fino al pitch di presentazione.

Contenuti:

- Società, Imprenditori, Professionisti e Manager. Filiera Produttiva, Struttura Aziendale e Procedure. Le StartUp. Business Model, Value Propositione e USP - I Clienti. I Canali e la Relazione con il Cliente. Risorse, Attività e Partnership - Struttura di Costi e Flussi di Ricavi. Startup Business Model Design. SWOT Analysis. Business Model Innovativi - Selezione del Personale e Public Speaking - Nozioni Fiscali e Legali

Modalità di esame:

Progetto di gruppo ed esame scritto individuale.

Criteria di valutazione:

L'accertamento di profitto avverrà in due fasi: - consegna e presentazione di un progetto di gruppo, - esame individuale. Il compito verifica la preparazione sulle basi teoriche presentate durante il corso, il progetto l'abilità di metterle in pratica su di un caso di studio concreto.

Testi di riferimento:

A. Osterwalder, Y. Pigneur, Business Model Generation. : Wiley, 2010 A. Baldissera, B. Bonaventura, Startup Marketing. : Francoangeli, 2013

Eventuali indicazioni sui materiali di studio:

Slide del corso messe a disposizione sul sito web del corso. La pagina Facebook del corso servirà per eventuali approfondimenti, aiuti e discussioni.

INFORMATION RETRIEVAL

Titolare: Prof. MASSIMO MELUCCI

Mutuato da: Percorso abilitante speciale in Scienze naturali, chimica e geografia, microbiologia

Periodo: Il anno, 2 trimestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 48A; 6,00

Sede dell'insegnamento: Dipartimento di Ingegneria dell'Informazione

Prerequisiti:

Fondamenti di informatica, calcolo delle probabilità e statistica.

Conoscenze e abilità da acquisire:

L'insegnamento si occupa di Information Retrieval e dei metodi e modelli per i motori di ricerca, nonché di argomenti più avanzati. Le lezioni, i compiti assegnati e il laboratorio hanno lo scopo di dare gli strumenti metodologici per il progetto e la realizzazione di funzionalità di information retrieval utili per applicazioni reali.

Attività di apprendimento previste e metodologie di insegnamento:

Lezione frontale ed attività di laboratorio.

Contenuti:

Gli argomenti principali necessari per la comprensione di un sistema di IR sono i seguenti: Metodi di indicizzazione e reperimento Modelli di reperimento Motori di ricerca Machine Learning Valutazione Altri argomenti sono ad esempio retroazione e metodi avanzati per il reperimento di informazione.

Modalità di esame:

Colloqui e presentazioni orali.

Criteria di valutazione:

Si terrà conto di eventuali relazioni di progetto oltre alla conoscenza e competenza della materia.

Testi di riferimento:

Massimo Melucci, Information Retrieval: metodi e modelli per i motori di ricerca. : Franco Angeli, 2013

Eventuali indicazioni sui materiali di studio:

Si veda il libro di testo.

INTELLIGENZA ARTIFICIALE

Titolare: Prof. ALESSANDRO SPERDUTI

Periodo: I anno, 2 trimestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 46A+14E; 8,00

Sede dell'insegnamento: Padova

Aule: Torre Archimede

Prerequisiti:

È opportuno avere familiarità con le conoscenze di base relative al Calcolo delle Probabilità e della Logica. Inoltre è consigliabile avere conoscenze di base relative alla Programmazione e agli Algoritmi. L'insegnamento non prevede propedeuticità.

Conoscenze e abilità da acquisire:

In questo insegnamento si presentano i concetti e le tecniche fondamentali di alcuni degli approcci principali, all'interno della Intelligenza Artificiale, per la soluzione di problemi difficili. In particolare sono esaminate tecniche di Ricerca in uno Spazio di Soluzioni, di Teoria dei Giochi, di Rappresentazione e Manipolazione di Conoscenza con e senza incertezza, di Pianificazione, e cenni di Sistemi con Vincoli ed Apprendimento Automatico. Al fine di sperimentare le difficoltà che tipicamente si incontrano nello sviluppare una applicazione di Intelligenza Artificiale, è previsto lo sviluppo da parte del singolo studente, o di un gruppo di studenti, di un piccolo progetto applicativo.

Attività di apprendimento previste e metodologie di insegnamento:

L'insegnamento prevede lezioni frontali.

Contenuti:

La struttura e le tematiche dell'insegnamento saranno le seguenti: - Introduzione, Motivazioni, Architetture di Agenti Intelligenti; - Risoluzione di Problemi e

Cenni di Sistemi con Vincoli; - Giochi con Avversario; - Rappresentazione della Conoscenza Tramite Logica Proposizionale e del Primo Ordine, Inferenza Logica; - Pianificazione; - Trattamento dell'Incertezza, Ragionamento Probabilistico; - Cenni di Apprendimento Automatico.

Modalità di esame:

Lo studente deve superare un esame scritto e, se ritenuto necessario dal docente, un esame orale. Inoltre lo studente deve sviluppare un piccolo progetto applicativo concordato con il docente.

Criteri di valutazione:

Il testo dell'esame scritto contiene alcune domande che consentono di valutare il livello di apprendimento delle nozioni impartite durante l'insegnamento e la capacità dello studente nell'analizzarle criticamente. Nel caso in cui la valutazione dello scritto non risulti soddisfacente per lo studente, il docente può integrare l'esame scritto con un esame orale per meglio verificare la preparazione dello studente. La valutazione del progetto considera la capacità, da parte dello studente, di individuare un caso di studio adeguato e di svolgere in modo autonomo un'attività di progettazione e realizzazione qualitativamente appropriata.

Testi di riferimento:

Stuart Russell, Peter Norvig, Artificial Intelligence: A Modern Approach. : Prentice Hall, 2010

Eventuali indicazioni sui materiali di studio:

Vengono rese disponibili, come riferimento, i lucidi utilizzati a lezione.

LINGUAGGI DI PROGRAMMAZIONE

Titolare: Prof. GILBERTO FILE'

Periodo: I anno, 1 trimestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 54A+18E; 10,00

Sede dell'insegnamento: Torre Archimede, Padova.

Prerequisiti:

Conoscenze approfondite dei linguaggi Java e C++.

Conoscenze e abilità da acquisire:

Conoscere un linguaggio funzionale (ML, Haskell). Apprezzare le differenze tra linguaggi funzionali ed imperativi. Apprezzare l'importanza dei tipi. Capire la gestione dei dati durante l'esecuzione di un programma (funzionale ed imperativo) e le sue implicazioni rispetto alla compilazione del linguaggio. Conoscere i temi principali che hanno segnato l'evoluzione dei linguaggi di programmazione dal 1950 a Java. La capacità di costruire un compilatore ed un interprete.

Attività di apprendimento previste e metodologie di insegnamento:

Il corso consiste fondamentalmente di lezioni tradizionali in aula. Per l'apprendimento è rilevante che ogni settimana una lezione di 2 ore sia organizzata come segue: nella prima ora gli studenti cercano di risolvere alcuni esercizi proposti dal docente sul materiale svolto nella settimana precedente. Nella seconda ora gli esercizi sono corretti alla lavagna con una forte interazione tra studenti e docente. Infine il progetto viene presentato agli studenti durante lo svolgimento del corso (in 5 parti) attraverso una documento ed alcune lezioni dedicate all'argomento. Inoltre il corso usa un sistema di elearning, basato sulla piattaforma Moodle, che consente un'interazione libera docente-studenti e anche studenti-studenti.

Contenuti:

I principali argomenti del corso sono i seguenti: 1) Un linguaggio funzionale (ML o Haskell): sintassi, esercizi, ricorsione, inferenza dei tipi, esecuzione eager e lazy; 2) Vari tipi di polimorfismo: parametrico, sovraccaricamento e di sottotipo; 3) Gestione run-time dei dati: blocchi, funzioni, ricorsione, scoping statico e dinamico, eccezioni; 4) Breve storia dei linguaggi orientati agli oggetti: Simula, Smalltalk, C++ e Java; 5) Pro e contro di C++; 6) Java a confronto con C++; 7) Il progetto consiste nella realizzazione di un compilatore per un semplice linguaggio funzionale: analisi lessicale, sintattica, generazione di codice intermedio, compilazione e interpretazione della traduzione finale.

Modalità di esame:

L'esame consiste di una prova scritta ed una orale. Nella prova scritta ci sono domande pratiche e domande teoriche. L'orale è una discussione sul progetto.

Criteri di valutazione:

La valutazione è una misura dell'assimilazione del materiale del corso da parte dello studente. Gli esercizi scritti pratici mostrano la capacità dello studente di applicare le nozioni apprese a problemi sempre diversi. Le domande teoriche mostrano la profondità e l'ampiezza dell'apprendimento dello studente. Per ultimo, l'esame orale mostra la comprensione da parte dello studente del progetto che mette in gioco diversi concetti rilevanti insegnati nel corso.

Testi di riferimento:

John Mitchell, Concepts in Programming Languages. : Cambridge University Press, 2003

Eventuali indicazioni sui materiali di studio:

Le slide usate a lezione sono tutte a disposizione degli studenti sul sito elearning del corso. Alcuni articoli, menzionati durante il corso, vengono resi disponibili sull'elearning. Lo stesso vale per gli esercizi svolti ogni settimana nella lezione speciale descritta in precedenza e per il documento del progetto. Anche esami passati vengono resi disponibili sul sito di elearning. Oltre a questo il corso segue un testo di riferimento.

LINGUAGGI DI PROGRAMMAZIONE AVANZATI

Titolare: Prof.ssa SILVIA CRAFA

Periodo: I anno, 3 trimestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 48A; 6,00

Sede dell'insegnamento: Padova

Prerequisiti:

Conoscenze di programmazione e di programmazione ad oggetti.

Conoscenze e abilità da acquisire:

Il corso presenta alcune tecniche avanzate dei moderni linguaggi di programmazione. Lo studente svilupperà la capacità di comprendere, ragionare e valutare alcune delle nuove tecniche di programmazione.

Attività di apprendimento previste e metodologie di insegnamento:

Lezioni frontali con esercizi ed approfondimenti di argomenti di ricerca tramite articoli scientifici.

Contenuti:

Il corso presenta alcune tecniche avanzate dei moderni linguaggi di programmazione, tra cui: l'uso dei sistemi di tipi per ragionare sui programmi, concetti avanzati di programmazione orientata agli oggetti (typing strutturale, type checking dinamico, mixins), linguaggi multi-paradigma, il design-by-contracts, programmazione concorrente basata sul modello ad attori. Tra i linguaggi su cui saranno affrontati questi argomenti ci sono Scala, C#, Spec#, Python, Ruby, Erlang, Go.

Modalità di esame:

Sono previste una prova scritta e una seconda prova che consiste nella discussione orale di un tema di approfondimento o in alternativa nella realizzazione di un progetto software.

Criteri di valutazione:

La prova scritta valuta l'acquisizione dello studente degli aspetti fondazionali affrontati durante il corso. La seconda prova valuta la capacità dello studente di analizzare e valutare aspetti avanzati dei linguaggi di programmazione.

Testi di riferimento:

M. Odersky, L. Spoon, B. Venners, Programming in Scala. : Artima, 2008 B.C. Pierce, Types and Programming Languages. : The MIT Press, 2002

LINGUAGGI E MODELLI PER IL GLOBAL COMPUTING

Titolare: Prof. PAOLO BALDAN

Periodo: I anno, 2 trimestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 48A; 6,00

Prerequisiti:

Il corso richiede familiarità con alcuni concetti matematici di base, quali relazioni, funzioni, insiemi, cardinalità, ordini parziali, principi di induzione, sistemi di deduzione basati su regole di inferenza. Sono utili alcune conoscenze di semantica dei linguaggi di programmazione. Il corso non ha propedeuticità.

Conoscenze e abilità da acquisire:

L'enorme diffusione dei sistemi concorrenti, distribuiti e mobili rende inadeguati i paradigmi di specifica e programmazione classici ed apre sfide complesse e affascinanti. Appare necessario un ripensamento, che parta dalle stesse fondamenta e che adotti un approccio rigoroso, formale, disciplinato. Il corso si propone di avvicinare lo studente a tematiche di interesse in questo ambito, utilizzando come strumenti sistemi di tipi, calcoli di processo e in generale linguaggi di modellazione. Parte da argomenti fondazionali oramai classici (come il Calculus of Communicating Systems ed il pi-calculus) e giunge ad illustrare alcuni argomenti di punta della ricerca nell'area. Vengono discussi alcuni linguaggi che traducono in pratica gli sviluppi teorici descritti, quali linguaggi evoluti per la concorrenza (Google Go, Erlang), linguaggi di orchestrazione (ORC) e linguaggi per programmazione service oriented (Jolie).

Attività di apprendimento previste e metodologie di insegnamento:

Lezioni in classe e uso di strumenti di verifica automatica.

Contenuti:

La struttura e le tematiche del corso saranno le seguenti: - Introduzione alla concorrenza e mobilità?: dagli automi ai sistemi reattivi e concorrenti. - Calculus of Communicating Systems (CCS), un linguaggio minimale per la descrizione di sistemi concorrenti. Equivalenza di processi: Sistemi di transizione e bisimulazione. - Logica di Hennessy-Milner e strumenti per la verifica. Mutua esclusione, deadlock, fairness. Proprietà di safety e liveness. - Verifica di proprietà con strumenti automatici. Il Concurrency Workbench ed il Mobility Workbench. - Sistemi con topologia dinamica e mobilità: pi-calcolo. Specifica di proprietà spaziali e cenni di applicazioni alla sicurezza dei protocolli. - Dai linguaggi di specifica ai linguaggi di programmazione: linguaggi avanzati per la concorrenza (Google Go, Erlang), linguaggi di orchestrazione (ORC) e linguaggi per programmazione orientata ai servizi (Jolie).

Modalità di esame:

Esercizi in classe, soluzione e discussione orale di esercizi avanzati, presentazione di un tema scelto dallo studente. Tra le opzioni ci sarà anche la realizzazione di un piccolo progetto che usi uno strumento di verifica.

Criteri di valutazione:

Lo studente è valutato rispetto alla sua capacità di risolvere semplici esercizi, verificando così l'acquisizione di nozioni e tecniche discusse durante il corso. Alcuni esercizi avanzati sono finalizzati a verificare la capacità di mettere a frutto quanto appreso per la soluzione di problemi nuovi. La presentazione verifica l'abilità dello studente di approfondire, autonomamente, tematiche di ricerca nell'area di interesse per il corso, e di esporre in modo efficace quanto appreso.

Testi di riferimento:

R. Milner, Communication and Concurrency. : Prentice Hall, 1989 L. Aceto, A. Ingólfssdóttir, K.G. Larsen, J. Srba, Reactive systems. : Cambridge University Press, 2007

Eventuali indicazioni sui materiali di studio:

Il libro di testo è complementato con articoli di ricerca e altre risorse disponibili online. Pagina web: <http://www.math.unipd.it/~baldan/Global>

LOGICA 2

Titolare: Prof. GIOVANNI SAMBIN

Mutuato da: Laurea magistrale in Matematica (Ord. 2011)

Periodo: I anno, 1 trimestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 32A+16E; 6,00

Prerequisiti:

E' caldamente suggerito, ma non strettamente necessario, aver seguito un corso di introduzione alla logica matematica.

Conoscenze e abilità da acquisire:

Potenzialità e limiti teorici del concetto di 'calcolabilità' e di metodo assiomatico. Possibilità che il calcolatore operi come assistente alla dimostrazione matematica, se fornito della fondazione appropriata.

Attività di apprendimento previste e metodologie di insegnamento:

Si intende sollecitare la partecipazione attiva di ogni studente, allo scopo di mettere in moto la sua visione critica, oltre che l'apprendimento nozionistico. Quindi le lezioni tradizionali saranno accompagnate da discussioni in aula e anche seminari su temi specifici svolti dagli studenti.

Contenuti:

Teoria della calcolabilità e teoremi di incompletezza. Più in dettaglio: Spiegazione informale della nozione di funzione calcolabile. Macchine di Turing, macchine a registri, abaci, funzioni ricorsive. Loro equivalenza e tesi di Church. Insiemi decidibili e ricorsivamente enumerabili. Sistema formale HA per la teoria dei numeri. Rappresentazione delle funzioni ricorsive. Aritmetizzazione della sintassi. Condizioni di Bernays-Hilbert-Loeb. Primo e secondo teorema di incompletezza di Gödel. Indecidibilità della logica dei predicati. Conclusioni. Fondazioni della matematica adatte ad una formalizzazione assistita dal calcolatore.

Modalità di esame:

Esame scritto con 4-5 semplici esercizi. Eventuale seminario durante il corso.

Criteri di valutazione:

Capacità dello studente di utilizzare i concetti appresi durante il corso in modo personale. Capacità di svolgere alcuni semplici esercizi, come applicazione dei concetti appresi e delle loro principali proprietà.

Testi di riferimento:

CONTENUTO NON PRESENTE

Eventuali indicazioni sui materiali di studio:

Dispense fornite dal docente.

METODI E MODELLI PER L'OTTIMIZZAZIONE COMBINATORIA

Titolare: Prof. LUIGI DE GIOVANNI

Periodo: Il anno, 1 trimestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 32A+4E+12L; 6,00

Sede dell'insegnamento: Padova

Prerequisiti:

Elementi di ricerca operativa, elementi di programmazione lineare, elementi di base di programmazione.

Conoscenze e abilità da acquisire:

Uso di metodologie avanzate di supporto alle decisioni per la modellazione e la soluzione di problemi di ottimizzazione combinatoria. Il corso intende fornire strumenti matematici e algoritmici per la soluzione di problemi pratici di ottimizzazione con l'utilizzo dei pacchetti software e delle librerie di ottimizzazione più diffusi.

Attività di apprendimento previste e metodologie di insegnamento:

L'insegnamento prevede lezioni frontali, esercitazioni in laboratorio e discussione di esempi notevoli. Le esercitazioni in laboratorio consistono nell'implementazione di algoritmi di ottimizzazione combinatoria sia esatti (con l'uso di librerie di programmazione lineare intera) sia euristici).

Contenuti:

1. Approfondimenti e applicazioni di Programmazione Lineare e dualità: metodo del semplice primale-duale, tecniche di generazione di colonne, applicazioni a problemi di ottimizzazione su grafo. 2. Metodi avanzati di Programmazione Lineare Intera (PLI): Branch & Bound e tecniche di rilassamento, formulazioni alternative di modelli PLI, metodo dei piani di taglio e tecniche di Branch & Cut, applicazioni ad esempi notevoli: commesso viaggiatore, problemi di localizzazione, problemi di network design etc. 3. Meta-euristiche di Ottimizzazione Combinatoria: ricerca di vicini e varianti, algoritmi evolutivi. 4. Applicazione di metodi di modellazione e ottimizzazione su grafo. 5. Laboratori: utilizzo di software e librerie di ottimizzazione.

Modalità di esame:

Esame orale sui contenuti del corso. Realizzazione facoltativa di un progetto individuale sulla soluzione di un problema, reale o realistico, di ottimizzazione combinatoria (definizione del problema, modellazione, applicazione di un metodo di soluzione esatto e/o euristico).

Criteri di valutazione:

L'esame verifica il livello di apprendimento degli argomenti svolti e la capacità dello studente di applicarli per la soluzione di problemi reali di ottimizzazione combinatoria.

Testi di riferimento:

CONTENUTO NON PRESENTE

Eventuali indicazioni sui materiali di studio:

Dispense fornite dal docente. Articoli scientifici.

PROVA FINALE

Titolare: da definire

Periodo: Il anno, 3 trimestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: ; 36,00

RETI WIRELESS

Titolare: Prof. CLAUDIO ENRICO PALAZZI

Periodo: Il anno, 1 trimestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 36A+8L; 6,00

Prerequisiti:

Reti di Calcolatori

Conoscenze e abilità da acquisire:

Questo corso offre una panoramica delle problematiche inerenti sistemi e servizi basati su reti wireless. A questo scopo, sono analizzati i principali problemi e soluzioni protocollari disponibili per ambienti wireless. Inoltre, sono discussi la terminologia, il funzionamento e le possibili alternative allo stato dell'arte nelle comunicazioni wireless. Attraverso l'analisi dei servizi che possono essere offerti su tecnologia wireless, lo studente diventerà consapevole delle possibili evoluzioni ed utilizzi futuri dei sistemi wireless. Infine, il corso si conclude con alcune nozioni utili all'implementazione di un elaborato volto all'analisi e alla progettazione di protocolli/applicazioni wireless. This class offers an overview of issues related to systems and services on wireless networks. To this aim, we analyze the main issues and protocol solutions available for wireless environments. Moreover, we discuss the terminology, the functioning and possible alternatives regarding the state-of-the-art in wireless communication. Through the analysis of services that can be offered over wireless technology, the student will become aware of the future possible evolution and utilization of wireless systems. Finally, the class ends with some notions useful to implement a project for the analysis and design of wireless protocols/applications.

Attività di apprendimento previste e metodologie di insegnamento:

L'insegnamento prevede lezioni frontali e la realizzazione di un progetto.

Contenuti:

Introduzione alle reti wireless. Problematiche relative alle reti wireless: perdite per errore e collisione, equità e ritardi di trasmissione, handoff Standard MAC: 802.11 a/b/g/n/p/s Protocolli di trasporto in ambiente wireless: TCP Vegas, TCP Westwood, TCP Hybla, CUBIC. Reti ad hoc e protocolli di routing: MANET, VANET, DSDV, AODV, DSR. Applicazioni e servizi su reti mobili.

Modalità di esame:

Gli studenti sono valutati attraverso progetti individuali o di squadra ed attraverso un esame orale sulle tematiche discusse in aula.

Criteri di valutazione:

L'esame orale finale e il progetto realizzato consentono di valutare il livello di apprendimento delle nozioni discusse in classe e l'abilità dello studente nel maneggiare concetti in modo pratico.

Testi di riferimento:

William Stallings, Wireless Communications & Networks (2nd Edition). : Prentice Hall, 2005

Eventuali indicazioni sui materiali di studio:

Vengono rese disponibili le trasparenze utilizzate in aula.

SICUREZZA

Titolare: Prof. MAURO CONTI

Periodo: I anno, 3 trimestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 40A; 6,00

Sede dell'insegnamento: Informazioni in lingua non trovate

Aule: Informazioni in lingua non trovate

Prerequisiti:

Conoscenze di base di sistemi distribuiti, crittografia e sicurezza delle reti.

Conoscenze e abilità da acquisire:

Acquisire conoscenze di sicurezza di sistema in ambiente Linux e Windows, sicurezza di rete wireless e wired, web-application security, sistema di gestione della sicurezza. Al termine del corso gli studenti saranno in grado di: progettare l'architettura di sistemi ed applicazioni sicure, e aggiornare autonomamente le proprie competenze nel settore.

Attività di apprendimento previste e metodologie di insegnamento:

Lezioni frontali, discussione di articoli scientifici.

Contenuti:

1) COMPUTER SECURITY TECHNOLOGY AND PRINCIPLES: Cryptographic Tools, User Authentication, Access Control, Database Security, Malicious Software, Denial-of-Service Attacks, Intrusion Detection, Firewalls and Intrusion Prevention Systems. 2) SOFTWARE SECURITY AND TRUSTED SYSTEMS: Buffer Overflow, Software Security, Operating System Security, Trusted Computing and Multilevel Security. 3) MANAGEMENT ISSUES: IT Security Management and Risk Assessment, IT Security Controls, Plans, and Procedures, Physical and Infrastructure Security, Human Resources Security, Security Auditing, Legal and Ethical Aspects. 4) PART FOUR CRYPTOGRAPHIC ALGORITHMS: Symmetric Encryption and Message Confidentiality, Public-Key Cryptography and Message Authentication. 5) NETWORK SECURITY: Internet Security Protocols and Standards, Internet Authentication Applications, Wireless Network Security.

Modalità di esame:

Scritta.

Criteri di valutazione:

Conoscenza dei concetti studiati nel corso.

Testi di riferimento:

W. Stallings, L. Brown, Computer Security: Principles and Practice 2/E. : Prentice Hall, 2011 M. Bishop, Introduction to Computer Security. : Addison-Wesley Professional, 2004

Eventuali indicazioni sui materiali di studio:

Libro (testo principale Computer Security: Principles and Practice 2/E) e articoli scientifici. Il corso sarà tenuto in Inglese. Il sito web del corso offrirà tutte le informazioni e materiale ulteriore: <http://www.math.unipd.it/~conti/teaching.html>

SISTEMI CON VINCOLI

Titolare: Prof.ssa FRANCESCA ROSSI

Periodo: I anno, 3 trimestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 38A+6E; 6,00

Sede dell'insegnamento: Torre Archimede, Via Trieste 63, Padova

Aule: Saranno pubblicate sul sito del corso di laurea.

Prerequisiti:

Nessuno.

Conoscenze e abilità da acquisire:

Argomento principale di questo corso e' la programmazione con vincoli, sia dal punto di vista teorico che pratico. La programmazione con vincoli e' un'area di ricerca molto attiva a cavallo tra l'Intelligenza Artificiale, la Ricerca Operativa, i Linguaggi di Programmazione, e le Basi di Dati, e fornisce strumenti per la modellazione e la soluzione di problemi reali visti come un insieme di vincoli su un certo insieme di variabili. Questi strumenti hanno molte applicazioni pratiche, dai turni del personale all'allocazione dei gate agli aerei, dalla schedulazione delle attivita' di un'azienda alla soluzione ottimizzata di problemi di logistica.

Attività di apprendimento previste e metodologie di insegnamento:

Lezioni ed esercitazioni in aula.

Contenuti:

* Introduzione al corso, esempi di problemi di vincoli. * Nozioni di base della programmazione con vincoli. * Alcuni risolutori completi. * Nozioni di consistenza locale. * Alcuni risolutori incompleti. * Algoritmi di propagazione di vincoli. * Metodi di ricerca nello spazio delle soluzioni. * Argomenti avanzati di programmazione con vincoli: o vincoli soft o vincoli bipolari o vincoli con incertezza

Modalità di esame:

Esame scritto piu' presentazione di un progetto svolto a gruppi.

Criteri di valutazione:

Lo scritto contiene alcune domande che consentono di valutare il livello di apprendimento delle nozioni impartite durante il corso. Il progetto permette agli studenti di approfondire alcune nozioni e di verificare il loro uso pratico in problemi simulati.

Testi di riferimento:

K. Apt, Principles of Constraint Programming. : Cambridge University Press, 2003 R. Dechter, Constraint processing. : Morgan Kaufmann, 2003 F. Rossi, P. Van Beek, T. Walsh, Handbook of Constraint Programming. : Elsevier, 2006

Eventuali indicazioni sui materiali di studio:

Verranno rese disponibili le trasparenze usate nelle lezioni.

SISTEMI CONCORRENTI E DISTRIBUITI

Titolare: Prof. TULLIO VARDANEGA

Periodo: I anno, 1 trimestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 52A+12E; 8,00

Sede dell'insegnamento: Torre Archimede

Aule: 2BC60

Prerequisiti:

L'insegnamento assume familiarità con l'architettura degli elaboratori tradizionali, con la struttura e le attività dei loro sistemi operativi, particolarmente per quanto attiene a concorrenza, sincronizzazione e gestione dell'I/O, e dei fondamenti delle reti. L'insegnamento non prevede propedeuticità.

Conoscenze e abilità da acquisire:

Il corso si propone di: - illustrare problematiche e modelli di base e avanzati di concorrenza (intesa come parallelismo potenziale) realizzata a software, studiando le soluzioni proposte da Java e Ada, in quanto linguaggi riccamente dotati di supporto diretto alla concorrenza, come strumenti di sperimentazione e di confronto; - analizzare i principi costruttivi e i paradigmi architetturali e realizzativi che stanno alla base dei sistemi distribuiti, nella loro evoluzione da sistemi multiprocessori omogenei a sistemi multicomputer eterogenei lascamente interconnessi.

Attività di apprendimento previste e metodologie di insegnamento:

Il corso si compone di due segmenti complementari. Nel primo segmento si prendono in esame modelli e paradigmi di programmazione concorrente, concentrandosi sulla concorrenza direttamente esprimibile a linguaggio (ossia senza ricorso a librerie esterne), utilizzando Java e Ada come linguaggi di sperimentazione. Nel secondo segmento si affronta invece l'evoluzione architetturale tecnologica dei sistemi distribuiti, culminando nell'analisi di CORBA come paradigma di interconnessione di sistemi eterogenei secondo il modello cliente-servernte. In questa parte del corso si illustrano anche i fondamenti di approcci particolarmente avanzati come virtualizzazione e cloud computing. Nell'ambito di entrambi i segmenti del corso, il docente propone allo studente esercizi da realizzare in proprio in laboratorio per sperimentare direttamente le problematiche progettuali e realizzative e i paradigmi di soluzione illustrati a lezione.

Contenuti:

Problematiche di concorrenza - Introduzione storica e metodologica - Nozione di processo e modalità di sincronizzazione - Un modello concreto e sue progressive estensioni - La dimensione temporale - Cenni sulla virtualizzazione Problematiche di distribuzione - Definizioni fondamentali - Comunicazione e sincronizzazione in distribuito - Il sistema dei nomi - Soluzioni concrete: Java RMI, Ada DSA, CORBA - La frontiera del cloud computing

Modalità di esame:

L'esame di profitto consiste nella redazione e nella discussione di una relazione scritta che illustri le problematiche affrontate nello svolgimento del progetto didattico assegnato dal docente, e le soluzioni adottate per risolverle. La presentazione della relazione viene accompagnata da una dimostrazione pratica del prodotto software realizzato in risposta ai requisiti del progetto.

Criteri di valutazione:

Lo sviluppo del progetto didattico viene accompagnato da intenso dialogo con il docente, che consente allo studente di approfondire le principali problematiche affrontate a lezione e associate alla realizzazione del progetto. La stesura della relazione tecnica mette alla prova la capacità di sintesi e di astrazione dello studente. La presentazione e discussione del progetto di fronte al docente consente di completare la valutazione il grado di apprendimento complessivo dello studente rispetto ai principali temi della materia.

Testi di riferimento:

Alan Burns and Andy Wellings, Concurrent and Real-Time Programming in Ada. : Cambridge University Press, 2007 Andrew S Tanenbaum, Maarten van Steen, Distributed Systems - Principles and paradigms. : Pearson Education International, 2006

Eventuali indicazioni sui materiali di studio:

Il docente pubblica regolarmente tutte le diapositive utilizzate a lezione e anche materiale supplementare utile per l'approfondimento dei temi trattati in aula.

SISTEMI IPERMEDIALI

Titolare: Prof.ssa OMBRETTA GAGGI

Periodo: I anno, 1 trimestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 44A; 6,00

Sede dell'insegnamento: Padova

Prerequisiti:

Sistemi Operativi, Reti e Sicurezza

Conoscenze e abilità da acquisire:

Il corso illustra le principali tecnologie per la codifica, memorizzazione e diffusione di informazioni multimediali, e analizza le applicazioni distribuite con

particolare riferimento all'ambiente Internet.

Attività di apprendimento previste e metodologie di insegnamento:

Lezioni frontali

Contenuti:

- Introduzione. Sistemi multimediali e ipermediali. I formati dei media. Media e modelli dei dati. Classificazione dei media. Audio, immagini statiche, video. Media statici, media continui, media temporizzati. - Le immagini. Rappresentazione digitale delle immagini. Risoluzione e profondità di colore. Percezione umana del colore. Modelli per la codifica dei colori. Tecniche di riduzione dei colori. Formati standard per la rappresentazione delle immagini: GIF, PNG, JPEG. Il formato JPEG2000. Le immagini vettoriali. - L'audio. Rappresentazione digitale delle informazioni audio. Campionamento e quantizzazione. Teorema di Nyquist. Rapporto segnale-rumore. Dimensione dei dati e banda di trasmissione. Formati standard per la codifica dell'audio: WAV, u-Law. I sistemi MIDI. - Il video. Rappresentazione del segnale video analogico. Standard NTSC e PAL. Il video digitale. Rappresentazione del colore. Sottocampionamento cromatico. Standard H261, H263, MPEG. - La compressione dei dati. Compressione reversibile e compressione irreversibile. Compressione entropica. Compressione LZW. Compressione dei dati acustici. Elementi di psicoacustica. Bande critiche. Mascheramento spaziale e temporale. Compressione MP3. Compressione JPEG delle immagini. Compressione video. Codifica predittiva. Vettori di movimento. Compressione MPEG. - La trasmissione dei dati continui. La suite di protocolli RTSP, RTCP e RTP. - Concetto di qualità di servizio (QoS) nella trasmissione di dati Multimediali: il protocollo RSVP, IntServ e DiffServ. - Architetture per la distribuzione di dati multimediali. Architetture client-server e P2P. Streaming e Jitter. Interazione tra flussi di rete elastici e real time, gestione del buffer. - I sistemi operativi per media continui. Gestione delle risorse. Qualità di servizio. Scheduling real-time. Algoritmi di scheduling per media continui. Cenni alla programmazione su SmartPhone.

Modalità di esame:

Esame orale o progetto

Criteri di valutazione:

L'esame verifica l'effettivo apprendimento dei concetti esposti durante l'insegnamento. Questo può avvenire in forma di discussione orale, oppure applicando quanto appreso nella progettazione e realizzazione di una applicazione per smartphone

Testi di riferimento:

Ze-Nian Li, Mark S Drew., Fundamentals of Multimedia. : Prentice Hall, 2004

Eventuali indicazioni sui materiali di studio:

Le slide del corso sono fornite sul sito web del corso

SISTEMI REAL-TIME

Titolare: Prof. TULLIO VARDANEGA

Periodo: I anno, 2 trimestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 36A+12E; 6,00

Prerequisiti:

L'insegnamento assume familiarità con l'architettura degli elaboratori tradizionali, con la struttura e le attività dei loro sistemi operativi, particolarmente per quanto attiene a concorrenza, sincronizzazione e gestione dell'I/O. L'insegnamento non prevede propedeuticità.

Conoscenze e abilità da acquisire:

Il corso si propone di esaminare la struttura dei sistemi software embedded soggetti a vincoli temporali, con l'obiettivo di evidenziarne le caratteristiche che più li differenziano dagli altri sistemi di calcolo. Attenzione sarà posta su alcuni paradigmi di progettazione e programmazione di tali sistemi, che ne facilitano l'analisi e la verifica.

Attività di apprendimento previste e metodologie di insegnamento:

Il corso esamina la struttura dei sistemi software embedded soggetti a vincoli di tempo reale, illustrando le principali problematiche nella loro progettazione, realizzazione e validazione. In particolare vengono affrontate: - caratterizzazione architetturale (livello hardware, software, e sistema) - controllo e gestione del tempo e delle interfacce hardware - progettazione e programmazione di software real-time - tecniche e approcci per la modellazione e l'analisi di sistemi real-time - problematiche di verifica e validazione. Nell'ambito del corso, il docente propone allo studente esercizi da realizzare in proprio in laboratorio per sperimentare direttamente le problematiche progettuali e realizzative e i paradigmi di soluzione illustrati a lezione, oltre a familiarizzare gli studenti con i più recenti sviluppi della teoria real-time intorno a tematiche di particolare interesse.

Contenuti:

- Introduzione: cenni storici e visione architetturale - Cenni sulla affidabilità e la tolleranza ai guasti - Il problema dell'ordinamento, tassonomia di algoritmi - Politiche di sincronizzazione nella gestione delle risorse condivise - Problematiche di sistema: una visione d'insieme della pila tecnologica - Estensione ai sistemi distribuiti - Estensione ai sistemi multiprocessore

Modalità di esame:

L'esame si svolge in una di due modalità a scelta dello studente. Una modalità richiede la redazione e la presentazione di una relazione tecnica sulle problematiche incontrate nell'adattamento a principi di progettazione e programmazione real-time di un piccolo sistema concorrente e distribuito individuato congiuntamente dallo studente e dal docente. L'altra modalità prevede lo studio critico e la presentazione di un lavoro di ricerca recente, che sviluppa qualcuno dei temi toccati in aula, scelto dallo studente tra un insieme di lavori individuati dal docente.

Criteri di valutazione:

Lo sviluppo della prova d'esame scelta dallo studente, indipendentemente dalle sue specifiche modalità, viene accompagnato da intenso dialogo con il docente, che consente allo studente di approfondire le principali problematiche affrontate a lezione e associate alla realizzazione del progetto. La presentazione e discussione da effettuare in sede d'esame consente di completare la valutazione il grado di apprendimento complessivo dello studente rispetto ai principali temi della materia.

Testi di riferimento:

Jane W.S. Liu, Real-Time Systems. : Prentice Hall, 2000

Eventuali indicazioni sui materiali di studio:

TECNOLOGIE OPEN-SOURCE

Titolare: Dott. FRANCESCO TAPPARO

Periodo: I anno, 3 trimestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 48A; 6,00

Prerequisiti:

Nessuno

Conoscenze e abilità da acquisire:

Conoscenza della storia del movimento open source e di tecnologie collaborative libere.

Attività di apprendimento previste e metodologie di insegnamento:

Lezioni frontali.

Contenuti:

Il corso si compone di due parti; nella prima si darà un'introduzione ai concetti ed alla storia del software libero ed open source, mentre nella seconda si introdurranno alcune tecnologie collaborative libere. I temi trattati saranno: - la cultura hacker del MIT - la nascita del progetto GNU - il movimento open source - Creative Common - RDF e ccrel - alcune tecnologie collaborative libere

Modalità di esame:

Orale

Criteri di valutazione:

Conoscenza degli argomenti impartiti a lezione; dimistichezza teorica e pratica con le tecnologie insegnate.

Testi di riferimento:

CONTENUTO NON PRESENTE

Eventuali indicazioni sui materiali di studio:

Slide e materiale indicato nelle slide quando necessario.

TECNOLOGIE WEB 2

Titolare: Prof. MASSIMO MARCHIORI

Periodo: I anno, 3 trimestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 48A; 6,00

Sede dell'insegnamento: Tipicamente, Torre Archimede.

Prerequisiti:

E' opportuno avere familiarità con gli elementi di base del web, così come forniti nel corso di "Tecnologie Web", in particolare HTML, CSS, XML, XSLT.

Conoscenze e abilità da acquisire:

L'obiettivo principale del corso è quello di dare una panoramica introduttiva di alcune tra le principali tecnologie web di livello avanzato, in modo da avere una visione ad alto livello del web attuale e del suo futuro.

Attività di apprendimento previste e metodologie di insegnamento:

L'insegnamento prevede lezioni frontali, con esempi illustrativi mostrati anche tramite connessione diretta al web.

Contenuti:

+ Web Usability Usabilità ed interazione con gli utenti, analisi multi-livello, come costruire un sito web di successo. + E-commerce Il caso studio dei siti di e-commerce, specializzazione dell'interazione col cliente. + Web Advertisement La pubblicità nei siti web, tecniche d'uso ed errori da evitare. + Web Search Web Site Search, Search Engine Optimization, testo ed ipertesto, il bene ed il male del web, i Social Information Systems. + Web Naming I nomi del web, loro usi ed abusi. + Il Web della Conoscenza Fondamenti del web semantico, rappresentazione della conoscenza, ontologie, semantic querying, syntactic querying, web reasoning.

Modalità di esame:

Lo studente deve superare uno scritto, e consegnare un progetto. Sopra una certa soglia minima di punteggio lo studente può opzionalmente richiedere un ulteriore esame orale.

Criteri di valutazione:

Il criterio di valutazione principale è la comprensione delle tecnologie web mostrate durante il corso. Questo significa quindi conoscere il funzionamento, i punti deboli ed i punti di forza delle tecnologie, la loro interazione nel contesto.

Testi di riferimento:

CONTENUTO NON PRESENTE

Eventuali indicazioni sui materiali di studio:

Il materiale di studio per l'esame è fornito tramite il sito web del corso (<http://corsi.math.unipd.it/tecweb2/>), attraverso risorse online.

Curriculum: Curriculum Fondamenti dell'informatica

Curriculum: Curriculum Intelligenza Artificiale

Curriculum: Curriculum Sistemi

Curriculum: Linguaggi