

FACOLTÀ	Scienze Matematiche Fisiche e Naturali
ANNO ACCADEMICO	2016/2017
CORSO DI LAUREA	Matematica
INSEGNAMENTO	Algoritmi e Strutture Dati
TIPO DI ATTIVITÀ	Affine
AMBITO DISCIPLINARE	Affine
CODICE INSEGNAMENTO	16670
ARTICOLAZIONE IN MODULI	NO
SETTORI SCIENTIFICO DISCIPLINARI	INF/01
DOCENTE RESPONSABILE	
CFU	6
NUMERO DI ORE RISERVATE ALLO STUDIO PERSONALE	102
NUMERO DI ORE RISERVATE ALLE ATTIVITÀ DIDATTICHE ASSISTITE	48
PROPEDEUTICITÀ	
ANNO DI CORSO	III
SEDE DI SVOLGIMENTO DELLE LEZIONI	Vedi Orario CIM
ORGANIZZAZIONE DELLA DIDATTICA	Lezioni frontali / Lezioni laboratorio
MODALITÀ DI FREQUENZA	Facoltativa
METODI DI VALUTAZIONE	Prova Orale, Prova Scritta, Prova Pratica
TIPO DI VALUTAZIONE	Voto in trentesimi
PERIODO DELLE LEZIONI	II semestre
CALENDARIO DELLE ATTIVITÀ DIDATTICHE	Come da calendario disponibile presso www.cs.unipa.it
ORARIO DI RICEVIMENTO DEGLI STUDENTI	Martedì e Giovedì Ore 15:00-17:00

OBIETTIVI FORMATIVI DEL CORSO

Esporre lo studente a tecniche fondamentali di progetto ed analisi di algoritmi. In particolare, si copre tutto lo spettro delle strutture dati fondamentali e dei principali paradigmi algoritmici, con cenni allo studio di complessità computazionale di problemi intrattabili. Si trattano anche aspetti ingegneristici fondamentali per l'implementazione di algoritmi efficienti.

ORE	LEZIONI FRONTALI
2	NOZIONI INTRODUTTIVE Algoritmi e Strutture Dati. Nozioni introduttive per la soluzione algoritmica di un problema, diverse soluzioni per uno stesso problema. Il problema dell'efficienza di un algoritmo.
4	TECNICHE EMPIRICHE E MATEMATICHE PER L'ANALISI DI ALGORITMI Analisi degli algoritmi. Velocità di crescita delle funzioni. Ricorrenze Fondamentali e Master Theorem (enunciati ed applicazioni)

4	MODELLI di CALCOLO, COMPLESSITA' COMPUTAZIONALE E ALGORITMI Random Access Machines, Complessità Computazionale RAM, Macchine di Turing e relazione fra esse (definizioni ed enunciati).
4	ALGORITMI DI ORDINAMENTO Lower bound per gli algoritmi di ordinamento: caso pessimo (cenni). Principali algoritmi di ordinamento: il mergesort, l' heapsort. Casi speciali.
10	PARADIGMI PER IL PROGETTO DI ALGORITMI EFFICIENTI Divide et Conquer, Programmazioni Dinamica, Tecniche Greedy. Esempi: Ricerca Minimo e Massimo, Moltiplicazione d'interi, Moltiplicazione di Matrici; Mergesort. Prodotto di n matrici. Longest Common Subsequence, Riconoscimento Grammatiche Context Free. Algoritmi Greedy: Optimal Storage on Tapes. Il Problema dello Zaino (versione "greedy")
6	STRUTTURE DATI AVANZATE ED OPERAZIONI SU INSIEMI Alberi e loro rappresentazione. Visite su alberi. Operazioni Fondamentali su Insiemi. Union-find. Alberi di Ricerca Ottimi, Schemi di Alberi Bilanciati, Dizionari e Code a Priorità, Mergeable Heaps, Code Concatenabili.

	LEZIONI DI LABORATORIO ED ESERCITAZIONI
3	STRUTTURE DATI ASTRATTE IN C Pile, Code e loro implementazione in C mediante array e liste concatenate. Valutazione di un'espressione in forma postfissa mediante una pila e sua implementazione in C.
7	ALGORITMI DI SORTING IN C Implementazione di algoritmi di sorting in C (MergeSort, Heapsort, Sorting di Interi)
8	PARADIGMI DI PROGETTO DI ALGORITMI IN C Ricorsione. Divide et Impera: ricerca del minimo e del massimo, ricerca binaria e loro implementazione in C. Programmazione Dinamica: Distanza di Edit fra due stringhe. La massima sottosequenza comune.

TESTI CONSIGLIATI	<i>R. Sedgewick</i> – Algoritmi in C, Addison-Wesley. <i>T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein</i> - Introduzione agli Algoritmi e strutture dati, McGraw Hill. <i>A.V. Aho, J.E. Hopcroft, J.D. Ullman</i>, The Design and Analysis of Computer Algorithms, Addison Wesley. <i>C. Demetrescu, I. Finocchi, G.F. Italiano</i>, Algoritmi e Strutture Dati, McGraw-Hill.
--------------------------	--

RISULTATI DI APPRENDIMENTO ATTESI

Conoscenza e capacità di comprensione

Acquisizione degli strumenti di base per l'analisi ed il progetto di algoritmi. Capacità di utilizzare il linguaggio specifico proprio della disciplina.

Capacità di applicare conoscenza e comprensione

Capacità di sviluppare software basati su algoritmi efficienti per problemi elementari

Autonomia di giudizio

Essere in grado di valutare le implicazioni e i risultati degli studi algoritmici che segue e della complessità computazionale dei problemi ad essi associati.

Abilità comunicative

Capacità di esporre i risultati salienti degli studi algoritmici, anche ad un pubblico non esperto. Essere in grado di evidenziare le ricadute tecnologiche delle teorie studiate.

Capacità d'apprendimento

Capacità di aggiornamento con la consultazione di testi avanzati e pubblicazioni scientifiche propri del settore dell'algoritmica. Capacità di seguire, utilizzando le conoscenze acquisite nel corso, sia corsi di master di primo livello, che corsi di laurea magistrali.