

FACOLTÀ	Ingegneria
ANNO ACCADEMICO	2013/2014
CORSO DI LAUREA	Ingegneria Informatica e delle Telecomunicazioni
INSEGNAMENTO	Algoritmi e Strutture Dati
TIPO DI ATTIVITÀ	Caratterizzante
AMBITO DISCIPLINARE	Ingegneria informatica
CODICE INSEGNAMENTO	01175
ARTICOLAZIONE IN MODULI	NO
NUMERO MODULI	
SETTORI SCIENTIFICO DISCIPLINARI	ING-INF/05
DOCENTE RESPONSABILE	Da designare
CFU	6
NUMERO DI ORE RISERVATE ALLO STUDIO PERSONALE	90
NUMERO DI ORE RISERVATE ALLE ATTIVITÀ DIDATTICHE ASSISTITE	60
PROPEDEUTICITÀ	Calcolatori Elettronici, Programmazione
ANNO DI CORSO	Terzo
SEDE DI SVOLGIMENTO DELLE LEZIONI	Consultare il sito www.ingegneria.unipa.it
ORGANIZZAZIONE DELLA DIDATTICA	Lezioni frontali Esercitazioni in aula e nelle aule informatiche
MODALITÀ DI FREQUENZA	Facoltativa
METODI DI VALUTAZIONE	Prova Scritta e Orale
TIPO DI VALUTAZIONE	Voto in trentesimi
PERIODO DELLE LEZIONI	Consultare il sito www.ingegneria.unipa.it
CALENDARIO DELLE ATTIVITÀ	Consultare il sito www.ingegneria.unipa.it

DIDATTICHE	
ORARIO DI RICEVIMENTO DEGLI STUDENTI	Da definire

<p>RISULTATI DI APPRENDIMENTO ATTESI</p> <p>Conoscenza e capacità di comprensione</p> <p>Lo studente alla fine del corso acquisirà una buona conoscenza dei principali algoritmi e delle più importanti strutture dati utilizzate nella programmazione avanzata. Sarà in grado di analizzare e comprendere il codice sorgente dei principali algoritmi utilizzati per lo sviluppo del software.</p> <p>Capacità di applicare conoscenza e comprensione</p> <p>Lo studente sarà in grado di valutare le caratteristiche, i vantaggi e le limitazioni dei principali algoritmi e strutture dati. Sarà in grado di progettare, analizzare e valutare le soluzioni software a problemi di media complessità.</p> <p>Autonomia di giudizio</p> <p>Lo studente sarà in grado sia di effettuare l'analisi di un problema che di progettare, a partire da precise specifiche, una opportuna soluzione software. Sarà in grado di valutarne la qualità di una soluzione software in termini di semplicità, leggibilità, efficienza e possibilità di riutilizzo.</p> <p>Abilità comunicative</p> <p>Lo studente acquisirà la capacità di comunicare ed esprimere problematiche inerenti l'oggetto del corso. Sarà in grado di sostenere conversazioni su tematiche relative alla implementazioni software di algoritmi e strutture dati efficienti. Sarà in grado di utilizzare un linguaggio semplice e chiaro per la descrizione dei processi di analisi e di sintesi di soluzioni software a problemi di media complessità.</p> <p>Capacità d'apprendimento</p> <p>Lo studente dovrà sviluppare la capacità di apprendere i processi di analisi e di sintesi relativi alla codifica di algoritmi di media complessità e alla relativa implementazione di librerie e strumenti software</p>

<p>OBIETTIVI FORMATIVI</p> <p>Al termine del modulo lo studente conoscerà i concetti di base necessari alla analisi e alla applicazione di alcuni tra i più diffusi algoritmi, e avrà una conoscenza di strumenti di media complessità relativi alla progettazione di software avanzato e/o scientifico .</p>
--

ORE FRONTALI	LEZIONI FRONTALI

5	<p>Introduzione al concetto di algoritmo, complessità computazionale degli algoritmi. Semplici esempi di algoritmi per la gestione di vettori e matrici: Prodotto di matrici di numeri reali e complessi. Utilizzo di una matrice di interi per la simulazione di macchine a stati finiti. Algoritmi di ordinamento e calcolo della loro complessità computazionale. Implementazione di funzioni per la gestione di insiemi con operatori bitwise.</p>
5	<p>Liste semplici, liste circolari. Inserimento, ricerca e cancellazione di un elemento in una lista. Esempi di strutture dati elementari: pile e code. Code aperte ai due estremi, code con priorità. Implementazione di una coda con priorità con heap binario e con heap binomiale. Utilizzo di una pila per l'implementazione di un semplice interprete di espressioni matematiche in notazione polacca inversa.</p>
5	<p>Algoritmi sulle stringhe. Distanza di Levenshtein.</p> <p>Alberi binari di ricerca. Ordinamento, inserimento e ricerca di dati in un albero binario. Calcolo della complessità computazionale delle operazioni su alberi binari.</p>
5	<p>Funzioni di hashing e implementazione di una tabella hash con un vettore di liste semplici. Utilizzo di tabelle hash e alberi binari per l'implementazione di insiemi e di mappe associative semplici e multiple.</p>
5	<p>Concetto di grafo, grafi orientati e non orientati, matrice di incidenza. Strutture dati per l'implementazione di un grafo. Algoritmo di Dijkstra e di Bellman-Ford per trovare il percorso di costo minimo in un grafo. Problemi NP: percorso euleriano minimo in un grafo. Tecniche di rilassamento per la soluzione approssimata di problemi NP. Minimo albero di copertura.</p>
5	<p>Algoritmi di ottimizzazione: discesa lungo il gradiente, metodo del gradiente coniugato.</p> <p>Tecniche di compressione dell'informazione, tecniche di entropy encoding e algoritmo di Huffman.</p>
5	<p>Strutture dati per la geometria, algebra geometrica, concetto di multivettore. Prodotto geometrico.</p> <p>Esempi di algoritmi e strutture dati per l'implementazione delle principali operazioni dell'algebra geometrica. Algoritmi e strutture dati per la codifica di oggetti geometrici elementari (punti rette, piani) con l'utilizzo dell'algebra geometrica.</p>
5	<p>Introduzione alla analisi numerica. Integrazione numerica di funzioni. Integrazione con il metodo Monte Carlo. Integrazione di equazioni differenziali.</p>

	ESERCITAZIONI
20	Esercitazioni sulla sintesi, l'analisi e le applicazioni degli algoritmi studiati. Implementazione in linguaggio C e/o java degli algoritmi e delle soluzioni software studiate.
TESTI CONSIGLIATI	