



UNIVERSITÀ DEGLI STUDI DI PALERMO

DIPARTIMENTO	Ingegneria
ANNO ACCADEMICO OFFERTA	2016/2017
ANNO ACCADEMICO EROGAZIONE	2018/2019
CORSO DILAUREA	INGEGNERIA INFORMATICA E DELLE TELECOMUNICAZIONI
INSEGNAMENTO	ALGORITMI E STRUTTURE DATI
TIPO DI ATTIVITA'	B
AMBITO	50289-Ingegneria informatica
CODICE INSEGNAMENTO	01175
SETTORI SCIENTIFICO-DISCIPLINARI	ING-INF/05
DOCENTE RESPONSABILE	VASSALLO GIORGIO Ricercatore Univ. di PALERMO LO PRESTI LILIANA Professore Associato Univ. di PALERMO
ALTRI DOCENTI	
CFU	6
NUMERO DI ORE RISERVATE ALLO STUDIO PERSONALE	96
NUMERO DI ORE RISERVATE ALLA DIDATTICA ASSISTITA	54
PROPEDEUTICITA'	
MUTUAZIONI	
ANNO DI CORSO	3
PERIODO DELLE LEZIONI	2° semestre
MODALITA' DI FREQUENZA	Facoltativa
TIPO DI VALUTAZIONE	Voto in trentesimi
ORARIO DI RICEVIMENTO DEGLI STUDENTI	LO PRESTI LILIANA Martedì 16:00 17:00 VASSALLO GIORGIO Mercoledì 14:00 16:00 Edificio 6 - terzo piano.

DOCENTE: Prof. GIORGIO VASSALLO- Gruppo G1

PREREQUISITI	Buona conoscenza del linguaggio C
RISULTATI DI APPRENDIMENTO ATTESI	<p>Capacita' di applicare conoscenza e comprensione Lo studente sara' in grado di valutare le caratteristiche, i vantaggi e le limitazioni dei principali algoritmi e strutture dati. Sara' in grado di progettare, analizzare e valutare le soluzioni software a problemi di media complessita. Sara' anche in grado di sviluppare nuove soluzioni software, valutandone la qualita' in termini di semplicita, efficacia ed efficienza. Tale capacita' verra' valutata principalmente durante le ore di esercitazione.</p> <p>Autonomia di giudizio Lo studente sara' in grado sia di effettuare l'analisi di un problema che di progettare, a partire da precise specifiche, una opportuna soluzione software. Sara' in grado di valutarne la qualita' di una soluzione software in termini di semplicita, leggibilita, efficienza e possibilita' di riutilizzo. L'autonomia di giudizio verra' valutata esaminando le soluzioni proposte dagli studenti a problemi software di media complessita. Lo studente verra' incoraggiato inizialmente a trovare e valutare autonomamente soluzioni ai problemi posti, al fine di potere comprendere la qualita' e l'utilita' delle soluzioni proposte successivamente dal docente.</p> <p>Abilita' comunicative Lo studente acquisira' la capacita' di comunicare ed esprimere problematiche inerenti l'oggetto del corso. Sara' in grado di sostenere conversazioni su tematiche relative alla implementazioni software di algoritmi e strutture dati efficienti. Sara' 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 complessita. Il carattere interattivo delle lezioni dovra' permettere la valutazione e il miglioramento delle abilita' comunicative dello studente.</p> <p>Capacita' d'apprendimento Lo studente dovra' sviluppare la capacita' di apprendere i processi di analisi e di sintesi relativi alla codifica di algoritmi di media complessita' e alla relativa implementazione di librerie e strumenti software. Il grado di apprendimento sara' valutato non in base alla capacita' di memorizzare concetti specifici ma in base alla capacita' di ricostruire ex novo partendo dal minor numero possibile di idee generali di base soluzioni software ottimali.</p>
VALUTAZIONE DELL'APPRENDIMENTO	<p>Prova scritta consistente in :</p> <p>1) Domande a risposte aperte sugli argomenti trattati nel corso 2) Prove di programmazione consistenti nella stesura di codice sorgente in linguaggio C relativo agli argomenti trattati nel corso</p> <p>Prova orale consistente in: domande sulla prova scritta e su altri argomenti trattati nel corso</p>
OBIETTIVI FORMATIVI	Lo studente alla fine del corso acquisira' una buona conoscenza dei principali algoritmi e delle piu' importanti strutture dati utilizzate nella programmazione avanzata. Sara' in grado di analizzare e comprendere il codice sorgente dei principali algoritmi utilizzati per lo sviluppo del software. La capacita' di comprensione dello studente verra' valutata, dopo l'esposizione dei principali concetti, durante le lezioni frontali con un dialogo diretto con gli studenti.
ORGANIZZAZIONE DELLA DIDATTICA	Lezioni frontali ed esercitazioni
TESTI CONSIGLIATI	Luciano M. Baroni et al., Programmazione Scientifica, Pearson Education

PROGRAMMA

ORE	Lezioni
5	Introduzione al concetto di algoritmo, complessita' 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 complessita' computazionale. Implementazione di funzioni per la gestione di insiemi con operatori bitwise.
4	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 priorit�. Implementazione di una coda con priorit� 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.
4	Algoritmi sulle stringhe. Distanza di Levenshtein. Alberi binari di ricerca. Ordinamento, inserimento e ricerca di dati in un albero binario. Calcolo della complessita' computazionale delle operazioni su alberi binari.
5	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.
4	Algoritmi di ottimizzazione: discesa lungo il gradiente. Tecniche di compressione dell'informazione, tecniche di entropy encoding e algoritmo di Huffman.

PROGRAMMA

ORE	Lezioni
4	Strutture dati per la geometria, algebra geometrica, concetto di multivettore. Prodotto geometrico. Esempi di algoritmi e strutture dati per l'implementazione delle principali operazioni dell'algebra geometrica. Strutture dati per la codifica di oggetti geometrici elementari (punti rette, piani) con l'utilizzo dell'algebra geometrica.
4	Introduzione alla analisi numerica. Integrazione numerica di funzioni. Integrazione numerica di semplici equazioni differenziali.
ORE	Esercitazioni
10	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.
10	Implementazione in linguaggio C degli algoritmi e delle soluzioni software studiate del corso.