

<b>STRUTTURA</b>	Scuola Politecnica - DICGIM
<b>ANNO ACCADEMICO</b>	2016/2017
<b>CORSO DI LAUREA</b>	Ingegneria Informatica e delle Telecomunicazioni
<b>INSEGNAMENTO</b>	Algoritmi e Strutture Dati
<b>TIPO DI ATTIVITÀ</b>	Caratterizzante
<b>AMBITO DISCIPLINARE</b>	Ingegneria informatica
<b>CODICE INSEGNAMENTO</b>	01175
<b>ARTICOLAZIONE IN MODULI</b>	NO
<b>NUMERO MODULI</b>	
<b>SETTORI SCIENTIFICO DISCIPLINARI</b>	ING-INF/05
<b>DOCENTE RESPONSABILE</b>	GIORGIO Vassallo  Ricercatore  Università degli Studi di Palermo
<b>CFU</b>	6
<b>NUMERO DI ORE RISERVATE ALLO STUDIO PERSONALE</b>	90
<b>NUMERO DI ORE RISERVATE ALLE ATTIVITÀ DIDATTICHE ASSISTITE</b>	60
<b>PROPEDEUTICITÀ</b>	Calcolatori Elettronici, Programmazione
<b>ANNO DI CORSO</b>	Terzo
<b>SEDE DI SVOLGIMENTO DELLE LEZIONI</b>	Consultare il sito <a href="http://politecnica.unipa.it">politecnica.unipa.it</a>
<b>ORGANIZZAZIONE DELLA DIDATTICA</b>	Lezioni frontali  Esercitazioni in aula e nelle aule informatiche
<b>MODALITÀ DI FREQUENZA</b>	Facoltativa
<b>METODI DI VALUTAZIONE</b>	Prova Scritta e Orale
<b>TIPO DI VALUTAZIONE</b>	Voto in trentesimi

<b>PERIODO DELLE LEZIONI</b>	Consultare il sito <a href="http://politecnica.unipa.it">politecnica.unipa.it</a>
<b>CALENDARIO DELLE ATTIVITÀ DIDATTICHE</b>	Consultare il sito <a href="http://politecnica.unipa.it">politecnica.unipa.it</a>
<b>ORARIO DI RICEVIMENTO DEGLI STUDENTI</b>	Martedì ore 11 o per appuntamento

## **RISULTATI DI APPRENDIMENTO ATTESI**

### **Conoscenza e capacità di comprensione**

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.

### **Capacità di applicare conoscenza e comprensione**

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.

### **Autonomia di giudizio**

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.

### **Abilità comunicative**

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.

### **Capacità d'apprendimento**

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

## OBIETTIVI FORMATIVI

Al termine del modulo lo studente conoscerà i concetti di base necessari alla analisi e all'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.

ORE FRONTALI	LEZIONI FRONTALI
5	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.
5	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.
5	Algoritmi sulle stringhe. Distanza di Levenshtein.  Alberi binari di ricerca. Ordinamento, inserimento e ricerca di dati in un albero binario. Calcolo della complessità computazionale delle operazioni su alberi binari.
5	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.
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.
5	Algoritmi di ottimizzazione: discesa lungo il gradiente, metodo del gradiente coniugato.  Tecniche di compressione dell'informazione, tecniche di entropy encoding e algoritmo di Huffman.
5	Strutture dati per la geometria, algebra geometrica, concetto di multivettore.

	<p>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>
	<b>ESERCITAZIONI</b>
20	<p>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.</p>
<b>TESTI CONSIGLIATI</b>	<p>Luciano M. Baroni et al., Programmazione Scientifica, Pearson Education</p>

