



# UNIVERSITÀ DEGLI STUDI DI PALERMO

<b>DIPARTIMENTO</b>	Matematica e Informatica		
<b>ANNO ACCADEMICO OFFERTA</b>	2021/2022		
<b>ANNO ACCADEMICO EROGAZIONE</b>	2023/2024		
<b>CORSO DILAUREA</b>	INFORMATICA		
<b>INSEGNAMENTO</b>	LABORATORIO DI ALGORITMI		
<b>TIPO DI ATTIVITA'</b>	B		
<b>AMBITO</b>	50166-Discipline Informatiche		
<b>CODICE INSEGNAMENTO</b>	17611		
<b>SETTORI SCIENTIFICO-DISCIPLINARI</b>	INF/01		
<b>DOCENTE RESPONSABILE</b>	FLORES MANUELA	Ricercatore a tempo determinato	Univ. di PALERMO
<b>ALTRI DOCENTI</b>			
<b>CFU</b>	6		
<b>NUMERO DI ORE RISERVATE ALLO STUDIO PERSONALE</b>	86		
<b>NUMERO DI ORE RISERVATE ALLA DIDATTICA ASSISTITA</b>	64		
<b>PROPEDEUTICITA'</b>	05880 - PROGRAMMAZIONE E LABORATORIO C.I. 16670 - ALGORITMI E STRUTTURE DATI 16784 - SISTEMI OPERATIVI 16450 - ARCHITETTURE DEGLI ELABORATORI 16671 - INFORMATICA TEORICA		
<b>MUTUAZIONI</b>			
<b>ANNO DI CORSO</b>	3		
<b>PERIODO DELLE LEZIONI</b>	2° semestre		
<b>MODALITA' DI FREQUENZA</b>	Facoltativa		
<b>TIPO DI VALUTAZIONE</b>	Voto in trentesimi		
<b>ORARIO DI RICEVIMENTO DEGLI STUDENTI</b>	<b>FLORES MANUELA</b> Lunedì 09:00 12:00 Google Meet community.unipa.it previo appuntamento via email Giovedì 09:00 12:00 Google Meet community.unipa.it previo appuntamento via email		

DOCENTE: Prof.ssa MANUELA FLORES

<b>PREREQUISITI</b>	Lo studente deve conoscere gli argomenti dei corsi propedeutici.
<b>RISULTATI DI APPRENDIMENTO ATTESI</b>	<p>Conoscenza e capacita' di comprensione Il corso mira alla capacita' di progettazione e realizzazione di strutture dati e algoritmi efficienti per la soluzione di problemi complessi e affronta con rigore metodologico aspetti di progettazione e di ingegnerizzazione del codice, con particolare attenzione al modello computazionale utilizzato.</p> <p>Capacita' di applicare conoscenza e comprensione Il corso prevede la realizzazione di elaborati pratici consistenti nell'implementazione in C++ degli algoritmi e delle strutture dati proposte.</p> <p>Autonomia di giudizio Gli studenti sono guidati ad apprendere in maniera critica e responsabile tutto cio' che viene spiegato in aula e ad arricchire le proprie capacita' di giudizio attraverso lo studio del materiale didattico indicato o fornito dal docente e attraverso la realizzazione di elaborati pratici.</p> <p>Abilita' comunicative Attraverso l'interazione durante le lezioni e le attivita' di laboratorio previste, il corso tendera' a favorire lo sviluppo della capacita' di comunicare in modo chiaro ed esaustivo le proprie ragioni ed argomentare le proprie conclusioni.</p> <p>Capacita' d'apprendimento Attraverso approfondimenti e consultazione dei testi di riferimento, gli studenti apprenderanno il trasferimento delle metodologie algoritmiche illustrate in applicazioni reali.</p> <p>Competenze acquisite Implementare algoritmi numerici di base (ad es. algoritmi di ricerca, algoritmi di ordinamento, algoritmi fondamentali dei grafi, algoritmo di abbinamento delle stringhe) per risolvere un problema applicativo e selezionare l'algoritmo adeguato per un particolare contesto. Modellare un problema del mondo reale usando appropriate strategie su grafi (ad esempio, alberi, metodi di attraversamento per grafi e alberi, alberi di ricoprimento di un grafo)</p>
<b>VALUTAZIONE DELL'APPRENDIMENTO</b>	<p>Gli studenti dovranno affrontare una prova scritta in cui saranno somministrate 5 domande sugli argomenti affrontati durante il corso. E' prevista anche una prova pratica, che richiede la progettazione di gruppo e l'implementazione, mediante l'uso del linguaggio C++, di algoritmi e le opportune strutture dati per la soluzione di un problema assegnato. Con tale attivita' si intende verificare le competenze acquisite, ovvero la capacita' di applicare le conoscenze maturate durante il corso, la capacita' di interazione e di lavoro in gruppo. Durante la prova orale, verranno discusse la prova scritta e la prova pratica e ciascuno studente avra' modo di descrivere le soluzioni proposte mostrando le proprie capacita' di comunicazione e di uso del linguaggio proprio della disciplina. La valutazione della prova e' espressa in trentesimi in accordo alla tabella di valutazione descritta di seguito.</p> <p>Tabella di valutazione</p> <p>Valutazione: Eccellente Voto: 30 - 30 e lode</p> <p>Esito: Ottima conoscenza degli argomenti, ottima proprieta' di linguaggio, buona capacita' analitica, efficace capacita' di interazione e di lavoro in gruppo, lo studente e' in grado di applicare in modo proficuo le conoscenze per risolvere i problemi proposti. Valutazione: Molto buono Voto: 27-29</p> <p>Esito: Buona padronanza degli argomenti, piena proprieta' di linguaggio, buona capacita' di interazione e di lavoro in gruppo, lo studente e' in grado di applicare in modo adeguato le conoscenze per risolvere i problemi proposti. Valutazione: Buono Voto: 24-26</p> <p>Esito: Buona conoscenza di base dei principali argomenti, discreta proprieta' di linguaggio, discreta capacita' di lavoro in gruppo, con buona capacita' di applicare in modo autonomo le conoscenze alla soluzione dei problemi proposti. Valutazione: Soddisfacente Voto: 21-23</p> <p>Esito: Discreta conoscenza degli argomenti principali dell'insegnamento, soddisfacente proprieta' di linguaggio, discreta capacita' di saper applicare in modo autonomo le conoscenze acquisite. Valutazione: Sufficiente Voto: 18-20</p> <p>Esito: Sufficiente conoscenza di base degli argomenti principali dell'insegnamento e del linguaggio tecnico. Valutazione: Insufficiente Esito: Non possiede una conoscenza accettabile degli argomenti trattati nell'insegnamento.</p>
<b>OBIETTIVI FORMATIVI</b>	Obiettivo del corso e' quello di dare gli strumenti teorici e pratici per progettare e realizzare soluzioni efficienti a problemi complessi. A tale scopo, si e' scelto di svolgere l'attivita' di laboratorio tramite implementazioni in linguaggio C++. Vengono inoltre discusse alcune applicazioni reali in cui gli algoritmi trattati

	<p>vengono comunemente utilizzati. Gli studenti avranno anche modo di usare alcune conoscenze acquisite durante il loro percorso formativo, con lo scopo di concentrarsi in modo approfondito sui dettagli implementativi delle strutture dati complesse, sull'analisi della complessità e le tecniche per implementare in modo efficiente un algoritmo in relazione alle risorse di calcolo a disposizione.</p>
<b>ORGANIZZAZIONE DELLA DIDATTICA</b>	<p>La didattica è organizzata mediante lezioni frontali in aula e in laboratorio, in cui si prevede sia un trasferimento della conoscenza ma anche, con il supporto del mezzo informatico, la contestuale verifica dell'acquisizione delle competenze pratiche richieste. Per incrementare il coinvolgimento degli studenti si utilizzano anche metodologie di collaborative learning.</p>
<b>TESTI CONSIGLIATI</b>	<p>Testi di riferimento/Textbooks:            Data Structures and Algorithms in C++, Michael T. Goodrich, Roberto Tamassia, David M. Mount, Wiley, 2011. ISBN 978-0470383278 (Linguaggio C++, Alberi di ricerca, Ordinamento di stringhe/Language C++, Search Trees, Sorting strings)            Algorithms in C++, Parts 1-4: Fundamentals, Data Structure, Sorting, Searching [3rd ed.], Robert Sedgewick. Addison-Wesley Professional, 1998. ISBN 978-0201510591 (Search Trees)            Algorithms in C++ Part 5: Graph Algorithms, 3rd Edition. Robert Sedgewick. Addison-Wesley Professional, 2002. ISBN 978-0201361186 (Algoritmi su grafi/ Graph algorithms)            Introduzione agli algoritmi e strutture dati (3 ed). Cormen, Leiserson, Rivest, Stein. Mc Graw Hill, 2010. ISBN 978-8838665158 (Polinomi e FFT/Polynomials and the FFT)</p> <p>Testi di consultazione/Consultation textbooks:            The magic of Algorithms! Lectures on some algorithmic pearls. Paolo Ferragina, Università di Pisa. 2019            C++. Guida essenziale per programmatori di Bjarne Stroustrup, Pearson, 2015. ISBN 978-8865184493</p>

### PROGRAMMA

ORE	Lezioni
4	Algoritmi e loro implementazione: fase progettuale, fase realizzativa. Analisi di algoritmi, modelli computazionali e livelli di memoria.
3	Problema dell'ordinamento in un modello a 2-livelli di memoria: complessità I/O e lower bound.
6	Implementazioni efficienti di dizionari. Alberi di ricerca. Esempio di struttura dati cache oblivious: B-alberi.
8	Strutture dati per rappresentare grafi: matrici di adiacenza e liste di adiacenza. Implementazioni efficienti di algoritmi su grafi: problema del minimo albero ricoprente (MST). Union find e code a priorità. Algoritmo di Kruskal, Algoritmo di Prim: implementazioni e valutazioni sperimentali. Algoritmi per la computazione del MST in memoria semiesterna. Algoritmi per i cammini minimi in un grafo orientato.
5	Il problema dell'ordinamento di stringhe. Radix sort. Multi-key Quicksort. Ordinamento di stringhe in un modello I/O.
4	Strutture dati per la memorizzazione e la ricerca di stringhe. Il problema della ricerca di pattern in stringhe.
2	Operazioni su polinomi. Algoritmi per la moltiplicazione veloce di polinomi.
ORE	Laboratori
6	Introduzione all'uso del linguaggio C++ orientato agli oggetti
6	Implementazioni di alberi di ricerca
6	Implementazione di algoritmi di sorting multi-way
4	Implementazione di grafi e visite di grafi
4	Implementazione di algoritmi per trovare il minimo albero ricoprente
4	Implementazione di algoritmi e strutture dati per l'ordinamento e la ricerca di stringhe
2	Implementazione di algoritmi per la moltiplicazione di polinomi. Uso della FFT.