

Denominazione	ALGORITMI E STRUTTURE DATI
Moduli componenti	-
Settore scientifico-disciplinare	IINF-05/A (ex ING-INF/05)
Anno di corso e semestre di erogazione	2° anno, 2° semestre
Lingua di insegnamento	Italiano
Carico didattico in crediti formativi universitari	6
Numero di ore di attività didattica assistita complessive e ripartite tra DE e DI	48 (pari a 6 CFU di Didattica Erogativa) + 6 ore di Didattica Interattiva
Docente	Non definito
Risultati di apprendimento specifici	<p>Il corso mira a fornire agli studenti una solida comprensione dei costrutti base per valutare la qualità degli algoritmi e delle strutture dati, nonché le tecniche avanzate per lo sviluppo di codice efficiente in linguaggio Python, sfruttando le conoscenze già acquisite sulla programmazione orientata agli oggetti.</p> <p><i>Conoscenze e comprensione.</i> Lo studente acquisirà conoscenze relative a: tipi di dato astratto (pile, code, alberi) e loro implementazioni; grafi e loro rappresentazione; algoritmi di visita (descrizione, implementazione e costo); tecniche di sorting e searching; algoritmi di base su grafi: cammini minimi, alberi di copertura minima.</p> <p><i>Capacità di applicare conoscenze e comprensione.</i> Lo studente svilupperà competenze per essere in grado di: scegliere le strutture dati per risolvere efficientemente un problema; progettare algoritmi e utilizzare le strutture dati per risolvere problemi di diversa complessità; analizzare l'efficienza delle strutture dati e degli algoritmi proposti.</p> <p><i>Autonomia di giudizio e pensiero critico:</i> Al termine del corso lo studente sarà in grado di gestire problematiche relative alla formulazione di algoritmi per la soluzione dei problemi posti e la loro implementazione nel linguaggio di programmazione di riferimento. Saprà inoltre comunicare con chiarezza le metodologie utilizzate e i risultati ottenuti.</p> <p><i>Abilità comunicative:</i> Al termine del corso lo studente sarà in grado di documentare e presentare con chiarezza le soluzioni algoritmiche sviluppate; comunicare efficacemente i risultati dell'analisi degli algoritmi e delle strutture dati attraverso relazioni scritte e presentazioni orali; partecipare attivamente a discussioni tecniche e lavori di gruppo, esponendo idee e soluzioni in modo chiaro e comprensibile.</p> <p><i>Capacità di apprendere:</i> Al termine del corso lo studente sarà in grado di aggiornarsi continuamente sulle nuove tecniche e metodologie di sviluppo di algoritmi e strutture dati; avrà sviluppato competenze di autoapprendimento per adattarsi a nuovi paradigmi e linguaggi di programmazione; saprà utilizzare risorse didattiche avanzate e documentazione tecnica per risolvere problemi complessi in modo autonomo.</p>
Programma	<p>Il programma del corso è composto dai seguenti contenuti didattici:</p> <ul style="list-style-type: none"> • Strutture dati avanzate: pile, code, alberi binari, alberi bilanciati, heap.

	<ul style="list-style-type: none"> • Algoritmi di ordinamento: quicksort, mergesort, heapsort. • Algoritmi di ricerca: ricerca binaria, ricerca in alberi, hashing. • Grafi: rappresentazione, algoritmi di visita (DFS, BFS), cammini minimi (Dijkstra, Bellman-Ford), alberi di copertura minima (Prim, Kruskal). • Tecniche di analisi degli algoritmi: complessità temporale e spaziale, notazioni asintotiche (O, Ω, Θ).
Tipologie di attività didattiche previste e relative modalità di svolgimento	<p>L'insegnamento è strutturato in lezioni di didattica frontale, incoraggiando l'interazione e la partecipazione attiva degli studenti, ed esercitazioni, integrate con le lezioni e svolte con l'ausilio di un elaboratore. Saranno mostrati esempi di progettazione ed implementazione di applicativi basati su programmazione orientata agli oggetti.</p> <p>È previsto inoltre l'utilizzo di tecnologie digitali per l'erogazione delle lezioni e delle esercitazioni.</p>
Metodi e criteri di valutazione dell'apprendimento	<p>La valutazione dell'apprendimento (sia per studenti frequentanti che non frequentanti) consiste nello svolgimento di una prova scritta contenente quesiti teorici e pratici relativi ai contenuti del corso proposti attraverso domande a risposta multipla. Sarà richiesta inoltre la definizione e codifica di un algoritmo in linguaggio Python. La valutazione del quesito progettuale terrà conto della comprensione delle specifiche di progetto e della qualità implementativa della soluzione proposta.</p> <p>Sarà prevista inoltre una prova orale facoltativa. In questo caso, la prova scritta concorrerà alla composizione del voto finale, nella misura del 70%. Il restante 30% della valutazione si baserà sul colloquio orale finale. La valutazione del colloquio orale è espressa in trentesimi e terrà conto della proprietà di linguaggio, della capacità argomentativa, di analisi critica e di ragionamento.</p>
Criteri di misurazione dell'apprendimento e di attribuzione del voto finale	<p>La valutazione dell'apprendimento prevede l'attribuzione di un voto finale espresso in trentesimi.</p> <p>Il voto finale sarà determinato attraverso l'esame scritto sopra dettagliato. La lode sarà valutata per i soli studenti che abbiano raggiunto la valutazione complessiva di 30/30 analizzando la capacità di proporre soluzioni corrette ed efficienti nella risoluzione di problemi.</p>
Propedeuticità	<p>Fondamenti di Informatica Programmazione Object Oriented</p>
Materiale didattico utilizzato e materiale didattico consigliato	<ul style="list-style-type: none"> - Report, documenti e slides fornite dal docente - Data Structures and Algorithms in Python di Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser. ISBN: 9781118290279.