



UNIVERSITÀ DI PISA

ALGORITMI E STRUTTURE DATI

ANTONIO VIRDIS

Anno accademico **2023/24**
CdS **INGEGNERIA INFORMATICA**
Codice **756II**
CFU **6**

Moduli	Settore/i	Tipo	Ore	Docente/i
ALGORITMI E STRUTTURE DATI	ING-INF/05	LEZIONI	60	ANTONIO VIRDIS

Obiettivi di apprendimento

Conoscenze

Le conoscenze che lo studente dovrà acquisire riguardano la complessità computazionale degli algoritmi e alcuni algoritmi di base per la soluzione di problemi su diverse strutture dati (array, liste, alberi, grafi). Lo studente inoltre dovrà acquisire conoscenze relative a elementi avanzati di programmazione a oggetti nel linguaggio di programmazione c++ (funzioni e classi modello, ereditarietà, eccezioni).

Modalità di verifica delle conoscenze

Tramite una prova finale scritta che presenta quesiti relativi alle conoscenze, esercizi di analisi di programmi, esercizi di programmazione di algoritmi sulle strutture dati introdotte.

Capacità

Analisi della complessità degli algoritmi. Progettazione di algoritmi e loro implementazione in c++ sulle strutture dati presentate.

Modalità di verifica delle capacità

Mediante una prova pratica in laboratorio di programmazione che riguarda la progettazione di un algoritmo e la sua realizzazione nel linguaggio c++.

La capacità di programmazione viene ulteriormente verificata con la prova scritta (test a risposta multipla).

Comportamenti

Saranno acquisite sensibilità alle problematiche della progettazione e valutazione di algoritmi, e della loro programmazione.

Modalità di verifica dei comportamenti

Tramite le due prove di esame.

Prerequisiti (conoscenze iniziali)

Prerequisito obbligatorio di questo insegnamento è il superamento dell'unità didattica Fondamenti di Programmazione

Indicazioni metodologiche

Il corso sarà erogato secondo le modalità stabilite dall'ateneo. Sono previste sessioni di lezioni teoriche ed esercitazioni pratiche al computer. Si consiglia a ciascuno studente di fornirsi di laptop personale per seguire le attività pratiche.

Programma (contenuti dell'insegnamento)

Nozione di algoritmo.

Definizione di complessità computazionale (notazioni O-grande, Omega-grande e Theta-grande).

Complessità dei programmi iterativi. Principi e metodi di programmazione ricorsiva. Complessità dei programmi ricorsivi: relazioni di ricorrenza.

Strutture lineari : array e liste. Principali algoritmi di ricerca (lineare, binaria) e ordinamento (selection-sort, bubble-sort, quicksort, mergesort, heapsort, counting sort, radix sort)



UNIVERSITÀ DI PISA

Alberi binari: memorizzazione, visite e programmazione di semplici algoritmi.

Alberi generici: : memorizzazione, visite e programmazione di semplici algoritmi.

Alberi binari di ricerca.

Tipo di dato heap.

Metodo di ricerca Hash.

Metodologie di costruzione di algoritmi: divide et impera, programmazione dinamica, programmazione greedy.

Algoritmo per trovare la più lunga sottosequenza comune fra due sequenze.

Algoritmo di Huffman di compressione del codice.

Limiti inferiori: metodo per trovarli mediante gli alberi di decisione.

Grafi orientati e non orientati: definizione e memorizzazione. Visita in profondità

Algoritmo di Kruskal per trovare il minimo albero di copertura

Algoritmo di Dijkstra per trovare i cammini minimi da un nodo a tutti gli altri nodi.

Cenni alla NP-completezza: problemi intrattabili, le classi P e NP, riducibilità fra problemi, problemi NP-completi, problemi non risolvibili.

Nozioni avanzate di programmazione a oggetti in c++: funzioni e classi modello, ereditarietà semplice, gestione delle eccezioni.

Bibliografia e materiale didattico

Gli appunti del corso e le slide delle lezioni e dei laboratori saranno messi a disposizione sulla piattaforma Microsoft Teams.

Testi consigliati:

Camil Demetrescu, Irene Finocchi, Giuseppe F. Italiano «ALGORITMI E STRUTTURE DATI 2/ED»

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein «INTRODUZIONE AGLI ALGORITMI E STRUTTURE DATI 3/ED»

Indicazioni per non frequentanti

Tutto il materiale didattico sarà reso disponibile sulla piattaforma Microsoft Teams.

Il docente è a disposizione per ricevimenti prenotabili tramite email.

Modalità d'esame

L'esame è costituito da due prove obbligatorie:

- Test a risposta multipla
- Prova pratica di programmazione

Per poter accedere alla prova pratica di programmazione lo studente deve aver raggiunto la sufficienza al test. Le prove devono essere svolte nello stesso appello.

La valutazione finale sarà calcolata tenendo conto dei voti delle due prove.

Il non superamento della prova pratica di programmazione annulla la valutazione del test a risposta multipla.

Ultimo aggiornamento 23/10/2023 09:52