

COURSE SYLLABUS

Algoritmi e Strutture Dati

2526-1-E3102Q106

Obiettivi

Scopo del corso è insegnare allo studente come progettare, valutare e implementare algoritmi efficienti, utilizzando in modo opportuno le tecniche di programmazione e le strutture dati adeguate.

Conoscenza e capacità di comprensione

- Conoscenza dei concetti fondamentali relativi ai problemi computazionali
- Conoscenza delle tecniche base per dimostrare la correttezza di un algoritmo
- Conoscenza del funzionamento dei paradigmi di programmazione ricorsiva e divide-et-impera
- Conoscenza del funzionamento di algoritmi di ordinamento
- Conoscenza delle tecniche per valutare l'efficienza di algoritmi e strutture dati
- Conoscenza del funzionamento e delle caratteristiche di diverse strutture dati fondamentali

Conoscenza e capacità di comprensione applicate

- Capacità di analizzare la correttezza formale di un algoritmi iterativo e ricorsivo
- Capacità di applicazione dei paradigmi di programmazione ricorsiva e divide-et-impera per risolvere nuovi problemi computazionali
- Capacità di applicazione delle tecniche di valutazione dell'efficienza degli algoritmi iterativi, ricorsivi e divide-et-impera
- Capacità di simulare il funzionamento di algoritmi complessi e strutture dati fondamentali su istanze specifiche
- Capacità di utilizzare correttamente le strutture dati fondamentali per risolvere nuovi problemi computazionali

Autonomia di giudizio

- Capacità di scegliere l'algoritmo più efficiente e/o più indicato per risolvere nuovi problemi computazionali specifici

- Capacità di scegliere la struttura dati più efficiente e/o più indicato per risolvere nuovi problemi computazionali specifici

Abilità comunicative

- Saper utilizzare correttamente lo pseudocodice per presentare formalmente e in modo non ambiguo un algoritmo
- Saper comprendere e utilizzare il lessico corretto per comprendere o descrivere le caratteristiche di un algoritmo e di una struttura dati

Capacità di apprendere

- Essere in grado di ricercare e selezionare criticamente un nuovo algoritmo più indicato per risolvere un nuovo problema computazionale
- Essere in grado di ricercare e selezionare criticamente una nuova struttura dati più indicata per risolvere un nuovo problema computazionale

Contenuti sintetici

Metodologie di base per progettare algoritmi e analizzarne l'efficienza. Strutture dati fondamentali: definizioni e utilizzo

Programma esteso

- Introduzione: Algoritmo, problema, istanza.
- Analisi di algoritmi: Valutazione dei tempi di esecuzione, caso pessimo, ottimo e medio.
- Programmazione ricorsiva e approccio Divide-et-Impera: Mergesort e Quicksort.
- Valutazione del tempo di esecuzione di algoritmi ricorsivi: equazioni di ricorrenza.
- Altri algoritmi di ordinamento: ordinamento in tempo lineare.
- Strutture dati fondamentali: Array, liste, pile e code.
- Alberi Binari e Alberi Binari di Ricerca
- Heap e code con priorità. Heapsort.
- Grafi e loro rappresentazione in memoria.
- Algoritmi di visita dei grafi: BFS e DFS.

Prerequisiti

Nozioni base di programmazione

Modalità didattica

Lezioni, esercitazioni e approfondimenti sperimentali. Attività di studio individuali supportate da materiali didattici in E-learning.

Le lezioni sono tenute in italiano. Tutte le attività verranno svolte in presenza: 32 ore di lezioni frontali in modalità erogativa, 44 ore (20 di esercitazione e 24 di laboratorio) in modalità interattiva.

Materiale didattico

T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduzione agli Algoritmi e Strutture dati, Ed. Mc. Graw Hill

Materiale integrativo disponibile sul sito e-learning.

Periodo di erogazione dell'insegnamento

Secondo semestre

Modalità di verifica del profitto e valutazione

Prova scritta: la valutazione massima della prova scritta è 30/30. Tale prova consiste in:

- Esercizi che richiedono sviluppo di un algoritmo ricorsivo o iterativo per la soluzione di un problema assegnato
- Esercizi di simulazione relativi ad algoritmi presentati a lezione
- Domande aperte relative alle nozioni teoriche presentate a lezione

Le domande di teoria valgono, complessivamente, 8 punti

Possono essere assegnati fino a 2 punti aggiuntivi in caso di esercizi svolti particolarmente bene.

Prove parziali:

La prova scritta può essere sostituita da due prove parziali, che si tengono a metà corso e nel primo appello di giugno, a fine corso. Le prove parziali sono riservate alle matricole. Ogni prova è costituita da:

- Un esercizio che richiede sviluppo di un algoritmo per la soluzione di un problema assegnato
- Un esercizio di simulazione su input specifici degli algoritmi presentati a lezione
- Domande aperte relative alle nozioni teoriche presentate a lezione

Ogni prova parziale ha valutazione massima 15/15: il voto finale si ottiene sommando i voti delle due prove parziali. È prevista la possibilità di recuperare una sola delle due prove parziali (in caso di insufficienza, assenza o anche per migliorare il voto) nell'appello di luglio. Possono essere assegnati fino a 2 punti aggiuntivi (totali per le due prove) in caso di esercizi svolti particolarmente bene.

Orario di ricevimento

Su appuntamento

Sustainable Development Goals
