

CORSO DI ALGORITMI E STRUTTURE DATI

Prof. ROBERTO PIETRANTUONO

Programma dell'A.A. 2022/23

Obiettivi, argomenti, materiali didattici

OBIETTIVI FORMATIVI

Fornire gli strumenti necessari per la progettazione e l'analisi di algoritmi e strutture dati. Al termine del corso, gli studenti acquisiranno familiarità con una ampia varietà di strutture dati ed algoritmi noti che risolvono problemi di carattere fondamentale nello sviluppo delle applicazioni informatiche; saranno in grado di applicare e combinare le principali tecniche e paradigmi di progettazione, nonché strutture dati avanzate, per la sintesi di algoritmi corretti ed efficienti; saranno in grado di analizzare la correttezza e la complessità asintotica degli algoritmi.

ARGOMENTI

Concetti introduttivi: algoritmi e strutture dati, ricorsione, analisi e progettazione degli algoritmi.

PRIMA PARTE

Tecniche di analisi e strutture dati elementari.

Analisi di correttezza: invariante di ciclo, correttezza di algoritmi ricorsivi.

Analisi di complessità: analisi asintotica, notazioni O , Ω , Θ ; analisi di algoritmi ricorsivi.

Strutture dati elementari: dizionari; pile e code; code di priorità; liste; tabelle hash e stringhe; alberi binari di ricerca.

Tecniche di progettazione. Classificazione di problemi, caratteristiche della soluzione, tecniche di progettazione: [divide et impera](#); [ricerca combinatoriale](#); [metodi euristici](#); [programmazione dinamica](#).

DIVIDE et IMPERA

Problemi ed algoritmi comuni.

Ordinamento: *mergesort*, *heapsort*, *quicksort*, ordinamento in tempo lineare (*counting sort*, *radix sort*, *bucket sort*), mediane e statistiche d'ordine.

RICERCA COMBINATORIALE E METODI EURISTICI

Ricerca esaustiva, ricerca combinatoriale; *backtracking*, *pruning* della ricerca.

Metodi euristici, ricerca locale ed algoritmi golosi ("*greedy*"). Cenni ad algoritmi meta-euristici, strategie di *evolutionary computation* (cenni)

Problemi ed algoritmi comuni.

Problemi combinatoriali (costruzione di subset e permutazioni), copertura minima di un insieme. Codici di Huffman. Il problema della selezione di attività. Il problema dello *job scheduling*.

PROGRAMMAZIONE DINAMICA

Introduzione alla programmazione dinamica. Ricerca esaustiva vs. ricerca *greedy* vs. programmazione dinamica. Applicazioni.

Problemi ed algoritmi comuni.

Problemi di *pattern matching* e *string matching*, il problema della *longest increasing sequence*, *edit distance*. Fibonacci. Problemi di partizionamento. Problema dello zaino. Ulteriori esempi.

SECONDA PARTE

Tecniche di analisi e strutture dati avanzate.

Strutture dati e tecniche di analisi avanzate: alberi RB, B-Alberi, alberi auto-aggiustanti, *heap* di fibonacci, strutture dati per insiemi disgiunti. Grafi. Rappresentazione, esplorazione in ampiezza e profondità, ordinamento topologico. Applicazioni. Tecniche di analisi di algoritmi avanzate: analisi ammortizzata.

COMBINAZIONE DI TECNICHE

per la risoluzione di categorie comuni di problemi:

Problemi ed algoritmi comuni.

1. Problemi su stringhe ed insiemi: (*approximate*) *string matching*, copertura di insiemi (*set cover* e *set packing*), LCS, edit distance
2. Problemi di teoria dei numeri (es.: algoritmi DES ed RSA), problemi aritmetici ed algebrici.
3. Problemi combinatoriali: generazione di permutazioni, di sottoinsiemi, di partizioni.
4. Problemi su grafi: alberi di connessione minimi, ricerca di cammini minimi.

Cenni ad ulteriori tecniche per la progettazione.

- Multithreading e parallelismo: progettazione di algoritmi multithread, algoritmi paralleli e distribuiti. Esempi (Fibonacci, ordinamento).

Problemi intrattabili. Introduzione a problemi NP ed NP-completi. Soddisfacibilità, riducibilità. Esempi di problemi NP-completi.

Traduttori ed interpreti: analisi lessicale, analisi sintattica, analisi semantica, interpreti, strutture dati usate nei traduttori.

Parte Esercitativa: Prevalentemente in C.

MATERIALE DIDATTICO

LIBRO DI TESTO ADOTTATO

- 1) Cormen, Thomas, Charles Leiserson, Ronald Rivest, and Clifford Stein. Introduction to Algorithms. 3rd ed. MIT Press, 2009. ISBN: 9780262033848.

TRASPARENZE DALLE LEZIONI ED ESERCITAZIONI

Lezioni ed esercitazioni del corso disponibili sul sito web docenti di Ateneo:

<http://www.docenti.unina.it/roberto.pietrantuono> (area download).

LIBRI CONSIGLIATI

- 2) S. Skiena. The Algorithm Design Manual, 3rd ed, Springer, 2020. ISBN-13: 978-3030542559, ISBN-10: 3030542556.