



UNIVERSITÀ DI PISA

ALGORITMICA E LABORATORIO

ANNA BERNASCONI

Anno accademico 2016/17
CdS INFORMATICA
Codice 008AA
CFU 12

Moduli ALGORITMICA E LABORATORIO	Settore/i INF/01	Tipo LEZIONI	Ore 96	Docente/i ANNA BERNASCONI ANDREA MARINO ROSSANO VENTURINI
--	---------------------	-----------------	-----------	--

Obiettivi di apprendimento

Conoscenze

Cosa è un algoritmo e cosa è un modello di calcolo. Saper caratterizzare i dati da elaborare strutturandoli nel modo più opportuno al fine di agevolarne l'uso da parte degli algoritmi. Conoscere le strutture dati e le tecniche fondamentali per il progetto di algoritmi (di base). Conoscere le tecniche di valutazione della complessità e le limitazioni inerenti dei problemi da risolvere. Caratterizzare i problemi indecidibili ed esponenziali. Conoscere il linguaggio C e saper realizzare con esso gli algoritmi e le strutture dati viste in classe.

Modalità di verifica delle conoscenze

Lo studente sarà valutato sulla base della capacità dimostrata di discutere e mettere in pratica i principali argomenti presentati durante il corso. Metodi:

- Esame scritto finale
- Esame orale finale
- Dimostrazione pratica finale in laboratorio

Capacità

Saper progettare e realizzare in un linguaggio imperativo (es. C) algoritmi corretti ed efficienti attraverso l'uso di strutture dati e tecniche algoritmiche di base. Saper valutare la complessità di un algoritmo in tempo e spazio, al caso pessimo e al caso medio, e secondo la complessità ammortizzata. Saper valutare le limitazioni inerenti del calcolo.

Comportamenti

Lo studente saprà realizzare in un linguaggio imperativo (es. linguaggio C) gli algoritmi e le strutture dati viste in classe. Lo studente saprà inoltre stimare la bontà (leggi, efficienza in tempo e spazio) di un software sulla base delle caratteristiche progettuali salienti, prescindendo dunque dalla macchina, dal linguaggio di sviluppo o dal sistema operativo sul quale il software viene eseguito. Allo stesso modo, saprà stimare le prestazioni degli algoritmi prima di arrivare alla fase di programmazione, debugging e profiling, mediante l'uso di modelli matematici opportuni.

Prerequisiti (conoscenze iniziali)

Conoscenza di un linguaggio di programmazione imperativo, preferibilmente il linguaggio C (che utilizzeremo per le nostre esercitazioni di laboratorio). Dimestichezza con l'uso dei puntatori e della ricorsione.

Nozioni matematiche: proprietà e limiti di funzioni, sommatorie, numeri di Fibonacci, permutazioni, coefficienti binomiali, aritmetica modulare, numeri primi, metodi di dimostrazione (per induzione, per assurdo, per casi).

Prerequisiti per il Laboratorio:

- Conoscenza approfondita della programmazione C per ciò che concerne gli operatori (aritmetici e relazionali), il flusso del controllo (If-then-else, while, for), le funzioni, gli array, i puntatori, le stringhe e l'allocazione dinamica della memoria. Quindi i capitoli 1-5 del libro "Il Linguaggio C", B.W. Kernighan e D.M. Ritchie, Pearson-Prentice Hall, seconda edizione, 2008.

Indicazioni metodologiche

Lezioni: frontali

Attività didattiche:



UNIVERSITÀ DI PISA

- Frequenza delle lezioni
- Partecipazione alle discussioni
- Studio individuale
- Attività in laboratorio

Frequenza delle lezioni: consigliata

Metodi di insegnamento:

- lezioni
- laboratorio

Programma (contenuti dell'insegnamento)

L'obiettivo del corso è quello di introdurre strutture dati e tecniche algoritmiche (di base) che consentano allo studente la risoluzione di problemi su sequenze, liste, alberi e grafi, in modo efficiente in tempo e/o spazio. Si discuteranno inoltre alcune tecniche analitiche per la valutazione delle prestazioni degli algoritmi, o delle limitazioni inerenti del calcolo.

Il corso prevede una intensa attività di laboratorio che porterà gli studenti a sperimentare in linguaggio C le nozioni e le tecniche algoritmiche apprese in classe.

PROGRAMMA

1. Complessità computazionale: modello di calcolo, dimensione dell'input e dell'output, caso pessimo e caso medio.
2. Limiti del calcolo: albero di decisione, limiti superiori e inferiori.
3. Divide-et-impera, Relazioni di ricorrenza, Teorema fondamentale.
4. Algoritmi per sequenze statiche e dinamiche: ricerca e ordinamento.
5. Ordinamento basato su confronti: Insertion sort, Merge-sort, Quick-sort, Heap sort.
6. Ordinamento di interi: Counting sort, Radix Sort.
7. Ordinamento di stringhe: qsort-based.
8. Sottosequenza di somma massima.
9. Programmazione dinamica: LCS, Partizione e Zaino
10. Algoritmi randomizzati: Quicksort.
11. Generazione di combinazioni e permutazioni
12. Analisi ammortizzata: doubling di array, contatore binario, k ricerche.
13. Dizionari: Alberi bilanciati (Alberi 2-3), Tabelle hash (liste di trabocco e indirizzamento aperto).
14. Alberi: rappresentazione e visite.
15. Grafi I: rappresentazione e visite (DFS e BFS), DAG e ordinamento topologico.
16. Grafi II: Ciclo/cammino euleriano, ciclo hamiltoniano.
17. Breve introduzione a problemi computazionali, indecidibilità, e trattabilità (P, NP, NPC, EXP-TIME).

Bibliografia e materiale didattico

- **[CLRS]** T. Cormen, C. Leiserson, R. Rivest, C. Stein. *Introduzione agli algoritmi e strutture dati*. McGraw-Hill, Terza edizione, 2010.
- **[DFI]** C. Demetrescu, I. Finocchi, G. Italiano. *Algoritmi e strutture dati*. McGraw-Hill, Seconda edizione, 2008. [Solo pagine 161-166](#).
- **[CGGR]** P. Crescenzi, G. Gambosi, R. Grossi, G. Rossi. *Strutture di dati e algoritmi: progettazione, analisi e programmazione (seconda edizione)*. Pearson, 2012. [Solo pagine 87-96](#).
- **[BFL]** A. Bernasconi, P. Ferragina, F. Luccio. *Elementi di Crittografia*. Pisa University Press, 2015. Solo i capitoli 2 e 3.

Per il laboratorio, un testo fra:

- **[KR]** B.W. Kernighan, D.M. Ritchie. *Il Linguaggio C*, Pearson-Prentice Hall, seconda edizione, 2008.
- **[KP]** A. Kelley, I. Pohl. *C: Didattica e Programmazione*, Addison-Wesley, quarta edizione, 2004.

Modalità d'esame

L'esame consiste di tre prove:

- Una prova scritta con esercizi atti a valutare l'apprendimento delle nozioni teoriche e la capacità di "problem solving" dello studente. Tale prova viene valutata in trentesimi, e si intende superata se la valutazione è maggiore o uguale a 18.
- Una prova in laboratorio che verifica la capacità dello studente di realizzare in C gli algoritmi di base visti in classe, risolvendo semplici problemi su array, liste, alberi e grafi. Tale prova è da intendersi come un *test di idoneità*.
- Una prova orale sul programma del corso, la cui valutazione è in trentesimi e tiene in considerazione il risultato riportato dallo studente nella prova scritta.
- Le prove possono essere sostenute in appelli diversi.
- **LA PROVA ORALE E QUELLA DI LABORATORIO POSSONO ESSERE SOSTENUTE IN QUALUNQUE ORDINE E SOLO DOPO AVER SUPERATO LA PROVA SCRITTA.**
- Se la prova orale non viene superata, occorre ripetere soltanto quella.



UNIVERSITÀ DI PISA

- SE LA PROVA DI LABORATORIO NON VIENE SUPERATA PER DUE VOLTE CONSECUTIVE, OCCORRE RIPETERE TUTTE LE PROVE GIA' SOSTENUTE.
- Chiaramente la registrazione del voto di esame potrà essere effettuata soltanto dopo che tutte e tre prove sono state superate con successo.

Pagina web del corso

<http://didawiki.di.unipi.it/doku.php/informatica/all-b/start>

Altri riferimenti web

Sistema di Autovalutazione per il laboratorio: <http://algo1617.dijkstra.di.unipi.it/>

Ultimo aggiornamento 17/05/2017 17:57