

Programma del corso di **Algoritmi e Strutture Dati** *Prof.ssa Rossella Petreschi*

Il corso si propone di introdurre metodologie di progettazione e di analisi di algoritmi utili alla risoluzione di problemi che nascono in vari campi dell'informatica, nonché presentare algoritmi e strutture dati avanzati che trovano effettivo uso nella risoluzione efficiente di problemi applicativi di rilievo.

Programma in dettaglio AA.2012/2013

Richiami di concetti generali (2/Cap.1-3)

Introduzione al concetto di algoritmo e breve storia dello stesso. I concetti di finitezza, effettività e definitezza. Validazione e correttezza. Bontà di un algoritmo. L'importanza di sfruttare tutte le proprietà di un problema. Classi di complessità computazionale: algoritmi polinomiali ed esponenziali. Le costanti moltiplicative. La scelta della struttura dati opportuna. Confronto fra diversi algoritmi.

Metodologie di progettazione e di analisi

Le tecniche del Backtracking e del Branch and Bound (1/Cap.13; 2/Cap.2; 7/Par.3,4;Cap.11): Generazione dell'albero delle t-ple. Visita dell'albero delle t-ple con la tecnica del backtracking e del branch and bound. Il concetto di soluzione ammissibile e soluzione ottima. Vincoli espliciti ed impliciti. Come ridurre l'analisi sull'albero degli stati. Criteri di terminazione. Esempi di problemi da affrontare con la tecnica del backtracking: le 8 regine; il ciclo hamiltoniano; i sottoinsiemi a somma fissa. Esempi di problemi da affrontare con la tecnica del branch and bound: l'assegnamento; la bisaccia; il commesso viaggiatore.

Algoritmi di approssimazione (7/Cap.11): Problemi NP- hard. Rapporto fra la soluzione approssimata e la soluzione esatta. Accuratezza della approssimazione e rapporto di accuratezza. Il problema della bisaccia a variabili intere e a variabili reali. Algoritmi di approssimazione per il problema del commesso viaggiatore: *il vicino più vicino*; *2giri intorno all'albero*. Il problema del commesso viaggiatore a distanze euclidee. Sulla complessità del problema del commesso viaggiatore.

Analisi ammortizzata (3/Cap.17): Il metodo degli aggregati, il metodo degli accantonamenti, il metodo del potenziale. Applicazione dei tre diversi metodi al computo ammortizzato di una sequenza di incrementi su un contatore e di una sequenza di operazioni elementari su pile. Gestione di tabelle dinamiche: computo ammortizzato di una sequenza di operazioni di inserzione e cancellazione. Quando e come dimezzare e duplicare una tabella. La funzione potenziale relativa alle operazioni di contrazione ed espansione.

Strutture di dati evolute

Rappresentazione di alberi (4/Cap.4): Computo del numero degli alberi binari. Numeri di Catalan. Corrispondenza fra alberi binari, alberi ordinali e sequenze di parentesi. Rappresentazioni succinte ed implicite. Rappresentazione succinta di alberi binari: in ampiezza e tramite sequenze di parentesi bilanciate. Le funzioni Rank, Select e Match.

Alberi binari di ricerca (3/Cap.12,13; 5/Cap.6; 6/Cap.7; 7/Par.6.3): Analisi delle operazioni di inserimento e di cancellazione. Analisi nel caso medio della costruzione di un albero binario di ricerca. Alberi binari di ricerca di altezza logaritmica: alberi AVL e alberi Rossi/Neri. Ribilanciamenti tramite rotazioni. Alberi autoaggiustanti. L'operazione di splay. Analisi ammortizzata su alberi splay. Utilità degli alberi di ricerca bilanciati nell'implementazione di memorie cache.

Estensione di strutture dati (3/Cap.14,18; 7/Par.7.4): Alberi di intervalli. Alberi bilanciati per alberi di intervalli. Prova di correttezza della struttura presentata. Alberi: definizione e calcolo dell'altezza. Inserimento e cancellazione su B-alberi. Divisione e contrazione di nodi. Aumento e decremento dell'altezza.

Heap di Fibonacci (2/Cap.4; 3/Cap.19): Definizione delle operazioni basiche su un heap di Fibonacci: inserimento di un nodo; incremento e decremento del valore di un singolo nodo. Estrazione del minimo. Funzione potenziale; limitazione del grado massimo: teorema di correttezza. Unione di due heap di Fibonacci. L'algoritmo di Dijkstra e la struttura dati heap di Fibonacci.

Gestione di insiemi disgiunti (1/Par.4.3; 3/Cap.21; 5/Cap.9): Operazioni di union e find. Alberi quickunion e quickfind per rendere costante o l'operazione di union o l'operazione di find. Bilanciamento come regola per limitare l'altezza degli alberi quickunion e quickfind. Euristiche di compressione per ottimizzare il comportamento degli alberi quickunion e quickfind bilanciati: il partizionamento dei nodi e la funzione \log^* . Computo ammortizzato di una sequenza di operazioni union/find.

Algoritmi avanzati

Il problema del flusso nelle reti (3/Cap.26; 5/Cap.14): Definizione di rete di flusso. Flusso e quantità di flusso in una rete. Reti con sorgenti e pozzi multipli. Reti residue. Correttezza del metodo delle reti residue. Tagli in reti di flusso. Metodo dei cammini aumentanti. Massimo flusso/minimo taglio. L'algoritmo di Ford e Fulkerson. L'algoritmo di Edmonds e Karp. Lemma delle distanze. Prova della complessità computazionale dell'algoritmo di Edmonds e Karp.

Il problema dell'abbinamento (1/Cap.17; 6/Cap.12): Proprietà dell'abbinamento su grafi qualunque e su grafi bipartiti. Cammini alternanti e aumentanti. Teorema del matrimonio. Teorema di Konig e Egervary. Generazione dell'albero ungherese e abbinamento massimo in grafi bipartiti. Abbinamento Massimo in grafi bipartiti e reti di flusso. L'algoritmo di Edmonds e la contrazione dei germogli. Abbinamento massimo nei grafi qualunque. Esempi di applicazione del problema dell'abbinamento alla vita reale

Corrispondenza fra stringhe (3/Cap.32): Stringhe e spostamenti. Suffissi e prefissi. Algoritmo ingenuo di corrispondenza fra stringhe. Regola di Horner e classi di congruenza.

Algoritmo di Rabin-Karp e sua dipendenza dalla dimensione del *pattern*. Corrispondenza tra stringhe e automi a stati finiti. Correttezza dell'algoritmo stringhe/automi.

Grafi e planarità(8/Cap.3): Grafi planari. Formula di Eulero. Teorema di Kuratowski. Embedding di un grafo nel piano. St-numerazione di grafi biconnessi. Algoritmo lineare di Even e Tarjan per generare una st-numerazione. Le funzioni LOW, PATH. Rappresentazione a cespuglio di grafi planari. PQ-trees. Generazione di un disegno piano per grafi planari.

Bibliografia

1. M.H. Alsuwaiyel *Algorithms: Design Techniques and Analysis*,
World Scientific
2. G Ausiello, R. Petreschi *L'informatica Invisibile*,
Mondadori Sapienza
3. T.H. Cormen, C.E. Leiserson, R.L. Rivest, C.Stein *Introduzione agli algoritmi*,
Jackson Libri
4. P. Crescenzi, G. Gambosi, R. Grossi *Strutture di dati e algoritmi*,
Pearson, Addison Wesley
5. C. Demetrescu, I. Finocchi, G. Italiano *Algoritmi e strutture dati*,
McGraw-Hill
6. J.K Kingston *Algorithms and data structures*,
Addison Wesley
7. A. Levitin *The Design and Analysis of Algorithms*,
Addison Wesley
8. T.Nishizeki,N.Chiba *Planar graphs:theory and algorithms*,
North-Holland

