

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.2010/2011

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

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 (1/Cap.13; 2/Cap.2; 4/Cap.18; 6/Cap.11; 8/Cap.11)

Le tecniche del Backtracking e del Branch and Bound (1/Cap.13; 2/Cap.2; 8/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. Come ridurre l'analisi sull'albero delle t-ple. 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.

Analisi ammortizzata (4/Cap.18, 6/Cap.2): 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 (1/Cap.4, 3; 4/Cap.14/15/21/22; 5/Cap.4; 6/Cap.6/9; 7/Cap.7/10)

Heap di Fibonacci (1/Cap.4 e 4/Cap.21): 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; limitazione del grado massimo. Unione di due heap di Fibonacci. L'algoritmo di Dijkstra e la struttura dati heap di Fibonacci.

Alberi binari di ricerca (4/Cap.14-15, 6/Cap.6 e 7/Cap.7): 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*.

Rappresentazione di alberi (3 e 5/Cap.4): Computo del numero degli alberi binari. 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*. Utilizzo della rappresentazione succinta degli alberi per i documenti *XML*. Rappresentazione implicita di alberi etichettati: codice di Prufer. Codifica e decodifica efficiente del codice di Prufer. Deduzione della tomografia di una rete tramite codice di Prufer correlato con una matrice *Origine/Destinazione*.

Gestione di insiemi disgiunti (4/Cap.22; 6/Cap.9;7/Cap.10): 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*. Complessità dell'algoritmo di Kruskal per la ricerca di un albero ricoprente di costo minimo.

Algoritmi avanzati (1/Cap.17; 4/Cap.27/34/37; 6/Cap.14; 7/Cap.12; 8/Cap.7/11)

Il problema dell'abbinamento (1/Cap.17; 7/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. L'algoritmo di Edmonds e la contrazione dei germogli. Abbinamento massimo nei grafi qualunque. Esempi di applicazione del problema dell'abbinamento alla vita reale

Algoritmi di approssimazione (4/Cap.37; 8/Cap.11): Problemi NP- hard. Rapporto fra la soluzione approssimata e la soluzione esatta. Accuratezza della approssimazione e rapporto di accuratezza. Algoritmi di approssimazione per il problema del commesso viaggiatore e per il problema della bisaccia.

Il problema del flusso nelle reti (4/Cap.27; 6/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. L'algoritmo di Ford e Fulkerson. L'algoritmo di Edmonds e Karp. Prova della complessità computazionale dell'algoritmo di Edmonds e Karp. Esempi di reti di flusso tratti dalla vita reale.

Corrispondenza fra stringhe (4/Cap.34; 8/Cap.7): Stringhe e spostamenti. Suffissi e prefissi. Algoritmo ingenuo di corrispondenza fra stringhe. Regola di Horner e classi di congruenza. Algoritmo di Rabin-Karp. Corrispondenza tra stringhe con gli automi a stati finiti. Algoritmo di Knuth-Morris-Pratt.

Bibliografia

1. M.H. Alsuwaiyel *Algorithms: Design Techniques and Analysis*,
World Scientific
2. G Ausiello, R. Petreschi *L'informatica Invisibile*,
Mondadori Sapienza
3. S. Caminiti, R. Petreschi *Rappresentazione implicita di alberi etichettati*,
<http://twiki.di.uniroma1.it/pub/ASD/WebHome/prufer.pdf>
4. T.H. Cormen, C.E. Leiserson, R.L. Rivest *Introduzione agli algoritmi*,
Jackson Libri
5. P. Crescenzi, G. Gambosi, R. Grossi *Strutture di dati e algoritmi*,
Pearson, Addison Wesley
6. C. Demetrescu, I. Finocchi, G. Italiano *Algoritmi e strutture dati*,
McGraw-Hill
7. J.K Kingston *Algorithms and data structures*,
Addison Wesley
8. A. Levitin *The Design and Analysis of Algorithms*,
Addison Wesley