

Algoritmi Strutture Dati e Complessità

Academic Year 2023-2024

Contents

1	Obiettivi Generali del Corso	1
2	Argomenti principali e riferimenti	1
2.1	Parte I: Introduzione al corso e concetti fondamentali	1
2.2	Parte II: Ordinamento e Selezione	2
2.3	Parte III: Grafi e Strutture Dati Fondamentali	2
2.4	Parte IV: Tecniche Algoritmiche.	3

1 Obiettivi Generali del Corso

I principali obiettivi formativi del corso sono i seguenti:

- Introdurre strutture dati e algoritmi fondamentali, offrendo al contempo una panoramica delle loro applicazioni;
- Fornire agli studenti gli strumenti per comprendere a fondo il funzionamento degli algoritmi in relazione ai problemi per cui sono stati proposti;
- Fornire agli studenti gli strumenti per analizzare la correttezza e la complessità degli algoritmi e delle strutture dati fondamentali;
- Introdurre tecniche algoritmiche fondamentali, quali il metodo Divide et Impera e la Programmazione Dinamica;
- In generale, abituare gli studenti al Problem Solving, ossia al progetto di algoritmi efficienti per la soluzione di problemi computazionali di interesse nelle applicazioni.

2 Argomenti principali e riferimenti

2.1 Parte I: Introduzione al corso e concetti fondamentali

Esempi introduttivi e analisi degli algoritmi. Operazioni elementari e complessità degli algoritmi. Analisi di algoritmi e programmi: modello RAM a costi uniformi. Dimensione dell'input. Primi esempi di analisi della complessità di algoritmi e programmi. Analisi del caso peggiore e cenni all'analisi del caso medio. Oltre il caso peggiore: analisi aggregata del costo di sequenze di operazioni. **Riferimenti:** [1, Sez. 2.1 e 2.2. Sezione 16.1 e Sezione 16.4 fino a metà di pag. 418].

Analisi del costo degli algoritmi e notazione asintotica. Crescita delle funzioni. Notazioni O , Ω e Θ . Notazioni o e ω . Ulteriori esempi di analisi di algoritmi. **Riferimenti:** [1, Capitolo 3: Introduzione e Sezioni 3.1 - 3.3].

Il metodo Divide-et-Impera Primi esempi. Ricorrenze e analisi di algoritmi ricorsivi. Risolvere ricorrenze: metodo di sostituzione e uso di alberi di ricorsione con esempi. Cenni al metodo principale. **Riferimenti:** [1, Capitolo 4, Sezioni 4.1 - 4.5].

2.2 Parte II: Ordinamento e Selezione

Ordinamento e Merge-Sort. Il problema dell'ordinamento. Ordinamento basato sull'approccio Divide-et-Impera: Merge-Sort. Analisi della correttezza e del costo computazionale del Merge-Sort. **Riferimenti:** [1, Sezione 2.3].

Quicksort. Altri esempi del metodo Divide et Impera. Algoritmo di ordinamento Quicksort: analisi della correttezza e del costo computazionale. **Riferimenti:** [1, Capitolo 7].

Complessità dell'ordinamento. Limite inferiore al costo degli algoritmi di ordinamento basati su confronti: illustrazione e prova del risultato. Algoritmi non basati su confronti e ordinamento con costo lineare: Counting sort, Radix sort e Bucket sort **Riferimenti:** [1, Sezioni 8.1 - 8.4].

Il problema della selezione. Introduzione: trovare il minimo e il massimo. Algoritmi per risolvere il problema della selezione in tempo lineare. Selezione randomizzata con tempo atteso lineare e selezione con costo lineare nel caso peggiore. Cenni all'analisi del tempo atteso per la selezione randomizzata. Analisi della selezione con costo lineare nel caso peggiore. **Riferimenti:** [1, Capitolo 9, senza dimostrazione del Teorema 9.3],

2.3 Parte III: Grafi e Strutture Dati Fondamentali

Strutture dati: Heap e code di priorità. Operazioni e loro complessità desiderata. Applicazioni: Implementazione efficiente dell'algoritmo di Dijkstra, ordinamento Heapsort, calcolo dei primi k elementi. Implementazione di heap con array. Heapify. Ridimensionamento di heap realizzati con array. Analisi aggregata del costo di sequenze di operazioni su heap realizzati con array (v. riferimenti relativi al ridimensionamento di tabelle hash). **Riferimenti:** [1, Capitolo 6].

Introduzione ai Grafi. Il concetto di grafo. Grafi diretti e indiretti. Terminologia e proprietà di base. Rappresentazioni di grafi. Rappresentazione con liste di adiacenza e matrici di adiacenza e loro costi. **Riferimenti:** [1, Appendice B4, Sezione 20.1]

Attraversamento di grafi e sue applicazioni. Visita in ampiezza (Breadth-First-Search o BFS) e cammini minimi in grafi (non) diretti non pesati. Visita in profondità (Depth-First-Search o DFS) di grafi diretti e non diretti e sue proprietà. Classificazione degli archi. Ordinamento Topologico in grafi diretti: applicazioni e condizioni per l'esistenza di un ordinamento topologico. Algoritmi per il calcolo di ordinamenti topologici basati sulla visita in profondità. Componenti connesse e relazione di raggiungibilità. Componenti fortemente connesse in grafi diretti. Calcolo di componenti fortemente connesse usando la DFS: algoritmo lineare (detto di Kosaraju). **Riferimenti:** [1, Appendice B2 e Capitolo 20. Si possono omettere le dimostrazioni della Sez. 20.5 ad eccezione della prova del Lemma 20.13].

Cammini minimi e algoritmo di Dijkstra. Il problema dei cammini minimi da sorgente singola in grafi pesati. BFS e cammini minimi: perché non funziona in grafi pesati. Cammini minimi in grafi pesati: proprietà dei cammini minimi e del rilassamento. Algoritmo di Dijkstra: correttezza e costo computazionale. Algoritmo di Bellman-Ford: correttezza e costo computazionale. **Riferimenti:** [1, Capitolo 22, ad eccezione della Sezione 22.4].

Metodo Greedy e Alberi Minimi Ricoprenti. Il problema del minimo albero ricoprente. Algoritmo di Prim: analisi della correttezza e del costo computazionale. Versione efficiente basata su heap. Algoritmo di Kruskal. Versione efficiente dell'algoritmo di Kruskal usando la struttura dati Union-Find. Analisi della correttezza dell'algoritmo di Kruskal. **Riferimenti:** [1, Capitolo 21. Si veda anche capitolo 19 per versione efficiente dell'algoritmo di Kruskal basata su Union-find].

Strutture dati: tabelle hash. Operazioni e loro complessità desiderata. Applicazioni. Funzioni hash e implementazione di tabelle hash. Gestione delle collisioni: gestione mediante liste di trabocco e mediante indirizzamento aperto. Principali tecniche di indirizzamento aperto. Fattore di carico e ridimensionamento di tabelle hash: analisi aggregata del costo di sequenze di operazioni di inserimenti e cancellazioni su tabelle hash. **Riferimenti:** [1, Sezioni 11.1 - 11.4, senza le dimostrazioni dei Teoremi 11.4 e 11.8. Capitolo 16: Introduzione, Sezione 16.1, Sezione 16.4 fino a metà di pag. 418].

Strutture dati: alberi binari di ricerca. Array ordinati e alberi di ricerca. Operazioni principali su array ordinati e alberi binari di ricerca e loro costo. Implementazione di alberi binari di ricerca. Bilanciamento: mantenimento e operazioni di rotazione a seguito di inserimenti e cancellazioni. Cenni agli alberi di ricerca bilanciati: alberi rosso-neri e loro altezza. **Riferimenti:** [1, Capitolo 12 e Sezione 13.1]

Strutture dati: mantenimento di insiemi disgiunti. Operazioni su insiemi disgiunti. Rappresentazione di insiemi disgiunti con liste concatenate. Rappresentazione di insiemi disgiunti con foreste. **Riferimenti:** [1, Capitolo 19, Sezioni 19.1, 19.2 e 19.3]

2.4 Parte IV: Tecniche Algoritmiche.

Approccio Greedy (Goloso) al Progetto di Algoritmi. Il paradigma Greedy: esempi introduttivi. Codici e algoritmo di Huffman. **Riferimenti:** [1, Sezioni 15.1 - 15.3].

Programmazione Dinamica. Esempi introduttivi: alcuni problemi e progetto di algoritmi per la loro soluzione che applicano il paradigma della programmazione dinamica. Esempi più complessi: il problema della bisaccia. **Riferimenti:** [1, Capitolo 14: Introduzione, Sezioni 14.2 - 14.4], note ed esercizi svolti del Prof. Marchetti-Spaccamela presenti sul sito del corso.

Programmazione Dinamica e Cammini Minimi. Cammini minimi tra tutti i nodi e programmazione dinamica: l'algoritmo di Floyd-Warshall. Chiusura transitiva di un grafo. **Riferimenti:** [1, Capitolo 23: Introduzione e Sezioni 21.1 e 21.2] e note del Prof. Marchetti-Spaccamela presenti sul sito del corso.

References

- [1] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduzione agli Algoritmi e Strutture Dati - IV edizione*. McGraw Hill, Milano, Italia, 2023.