

# Prefazione

Prima dei calcolatori c'erano già gli algoritmi. Ma adesso che ci sono i calcolatori, ci sono più algoritmi e gli algoritmi sono il cuore del calcolo.

Questo libro è un'introduzione completa allo studio moderno degli algoritmi per calcolatori. Presenta numerosi algoritmi e li tratta molto approfonditamente, pur rendendone accessibili la progettazione e l'analisi ai lettori di qualsiasi livello. Abbiamo cercato di mantenere una semplicità espositiva, senza sacrificare l'approfondimento delle tematiche e il rigore matematico.

Ogni capitolo presenta una classe di algoritmi, le tecniche di progettazione, un'area di applicazioni e gli argomenti correlati. La descrizione degli algoritmi, sempre molto particolareggiata, si avvale di speciali "pseudocodici" appositamente progettati per essere leggibili da chiunque abbia un minimo di esperienza di programmazione. Il libro contiene 236 figure – molte composte da più parti – che illustrano il funzionamento degli algoritmi. Poiché l'*efficienza* è un criterio fondamentale della progettazione, abbiamo incluso l'analisi dei tempi di esecuzione di tutti i nostri algoritmi.

Il testo è stato ideato principalmente per essere utilizzato in corsi universitari o di specializzazione in algoritmi e strutture dati. Poiché si occupa di tecniche ingegneristiche di progettazione degli algoritmi, come pure di aspetti matematici, è adatto anche a professionisti ed esperti informatici.

In questa terza edizione abbiamo aggiornato di nuovo l'intero libro. Le modifiche riguardano vari aspetti, quali la revisione dello pseudocodice, l'inserimento di nuovi capitoli e uno stile di scrittura più attivo.

## **Ai docenti**

Questo libro è stato progettato per essere uno strumento versatile e completo. Potrà essere utilizzato in vari corsi, da quelli sulle strutture dati a quelli più avanzati sugli algoritmi. Poiché abbiamo incluso molto più materiale di quello che si trova in un tipico corso semestrale, non dovrete incontrare difficoltà a organizzare i vostri corsi utilizzando soltanto i capitoli che vi servono. Abbiamo cercato di rendere i capitoli relativamente autonomi, in modo che non dobbiate preoccuparvi per una imprevista e inutile dipendenza di un capitolo dall'altro. Ogni capitolo tratta prima gli argomenti più semplici e poi quelli più difficili, con la struttura dei paragrafi che marca i punti naturali di separazione. In un corso per studenti universitari, potreste utilizzare soltanto i primi paragrafi di un capitolo; in un corso di specializzazione, potreste trattare l'intero capitolo.

Abbiamo incluso 957 esercizi e 158 problemi. Ogni paragrafo termina con gli esercizi e ogni capitolo termina con i problemi. Generalmente, gli esercizi sono domande concise che verificano la padronanza degli argomenti trattati. Alcuni sono semplici esercizi di autoverifica, mentre altri sono più complessi e sono adatti per essere assegnati come compiti per casa. I problemi sono casi di studio più elaborati che spesso introducono nuovi argomenti; tipicamente, sono formati da più domande che guidano lo studente fino alla soluzione finale.

Sulla base della nostra esperienza con le precedenti edizioni di questo libro, abbiamo reso disponibili le soluzioni di alcuni problemi ed esercizi. Il nostro sito web <http://mitpress.mit.edu/algorithms/> contiene i link per queste soluzioni. Visi-

tate questo sito per verificare che esso non contenga la soluzione di un problema o esercizio che volete assegnare. Noi prevediamo di aumentare gradualmente nel tempo il numero delle soluzioni, quindi vi consigliamo di visitare il nostro sito prima di iniziare un corso.

I paragrafi e gli esercizi contraddistinti con una stella (★) sono più adatti agli studenti che frequentano corsi di specializzazione postlaurea. Un paragrafo con una stella non è necessariamente più difficile di uno senza stella, ma potrebbe richiedere conoscenze matematiche di livello elevato. Analogamente, gli esercizi con le stelle potrebbero richiedere un bagaglio di conoscenze più avanzate o una creatività superiore alla media.

### **Agli studenti**

Ci auguriamo che questo libro possa essere una piacevole introduzione allo studio degli algoritmi. Abbiamo cercato di rendere chiare e interessanti le descrizioni degli algoritmi. Per aiutarvi a capire gli algoritmi difficili o nuovi, abbiamo descritto i singoli passi di ogni algoritmo; abbiamo anche fornito spiegazioni accurate dei concetti matematici che sono richiesti per capire l'analisi degli algoritmi. Abbiamo organizzato i capitoli in modo che, se conoscete già qualche argomento, possiate sorvolare sui paragrafi introduttivi e passare rapidamente agli argomenti più avanzati. Il libro tratta numerosi argomenti. Abbiamo cercato di creare un libro che possa servirvi adesso come testo di riferimento di un corso e in futuro come manuale di consultazione nella vostra carriera professionale.

Quali sono i prerequisiti per leggere questo libro?

- Dovreste avere qualche esperienza di programmazione. In particolare, dovrete conoscere le procedure ricorsive e le strutture dati fondamentali, come gli array e le liste concatenate.
- Dovreste avere una certa dimestichezza con le dimostrazioni per induzione matematica. Poche porzioni del libro richiedono la conoscenza del calcolo infinitesimale. Al di là di questo, le Parti I e VIII del libro descrivono tutte le tecniche matematiche di cui avrete bisogno.

Abbiamo sentito, forte e chiara, la richiesta di fornire le soluzioni degli esercizi e dei problemi. Il nostro sito web <http://mitpress.mit.edu/algorithms/> contiene i link per le soluzioni di alcuni esercizi e problemi. Visitate il sito per confrontare le vostre soluzioni con le nostre. Vi chiediamo, però, di non inviarci le vostre soluzioni.

### **Ai professionisti**

La vasta gamma di argomenti trattati fa di questo libro un'eccellente guida agli algoritmi. Poiché ogni capitolo è relativamente autonomo, potrete focalizzare l'attenzione sugli argomenti che maggiormente vi interessano.

Quasi tutti gli algoritmi trattati sono di grande utilità pratica; di conseguenza abbiamo analizzato anche i problemi implementativi e altri aspetti della progettazione. Per quei pochi algoritmi che hanno soltanto un interesse squisitamente teorico abbiamo indicato le alternative pratiche.

Se vorrete implementare qualche algoritmo, sarà semplice tradurre il nostro pseudocodice nel vostro linguaggio di programmazione preferito. La struttura della pseudocodifica è stata studiata per presentare gli algoritmi in modo chiaro e conciso; di conseguenza, non considereremo la gestione degli errori e altri argomenti di ingegneria del software che richiedono ipotesi specifiche sul vostro particola-

re ambiente di programmazione. Abbiamo cercato di presentare ogni algoritmo in maniera semplice e diretta, evitando che le idiosincrasie di qualche particolare linguaggio di programmazione ne offuscino l'essenza.

Dal momento che voi utilizzate questo libro al di fuori di un corso, non avete la possibilità di confrontare le vostre soluzioni degli esercizi e dei problemi con quelle di un istruttore. Il nostro sito web <http://mitpress.mit.edu/algorithms/> contiene i link per le soluzioni di alcuni esercizi e problemi. Visitate il sito per confrontare le vostre soluzioni con le nostre. Vi chiediamo di non inviarci le vostre soluzioni.

### **Ai nostri colleghi**

Il libro include una ricca bibliografia e numerosi riferimenti alla letteratura corrente. Ogni capitolo termina con uno speciale paragrafo – Note – che fornisce alcuni dettagli storici e riferimenti bibliografici. Questo paragrafo non è una guida bibliografica completa per tutta la materia degli algoritmi. Sebbene possa sembrare difficile da credere per un libro di questa mole, non abbiamo potuto includere molti algoritmi interessanti per mancanza di spazio.

### **Novità della terza edizione**

Che cosa è cambiato tra la seconda e la terza edizione di questo libro? Il numero di modifiche pareggia quello delle modifiche tra la prima e la seconda edizione. Come abbiamo detto a proposito delle modifiche della seconda edizione, a seconda di come si guardano, si può dire che il libro non è cambiato molto o è cambiato un bel po'.

Da una rapido esame dell'indice generale si vede che la maggior parte dei capitoli e dei paragrafi della seconda edizione sono rimasti nella terza edizione. Abbiamo tolto due capitoli e un paragrafo, ma abbiamo aggiunto due nuovi capitoli, una nuova appendice e due nuovi paragrafi.

Abbiamo mantenuto l'organizzazione ibrida delle prime due edizioni. Anziché organizzare i capitoli soltanto in funzione dei problemi o soltanto in funzione delle tecniche, questo libro applica entrambi i criteri di organizzazione. Contiene capitoli basati sulle tecniche che trattano il metodo divide et impera, la programmazione dinamica, gli algoritmi golosi, l'analisi ammortizzata, la NP-completezza e gli algoritmi di approssimazione. Ma il libro include anche intere parti sull'ordinamento, sulle strutture dati per gli insiemi dinamici e sugli algoritmi per i problemi relativi ai grafi. Sappiamo che, sebbene voi abbiate bisogno di sapere come applicare le tecniche per progettare e analizzare gli algoritmi, i problemi raramente indicano quali siano le tecniche più appropriate per essere risolti.

Ecco una sintesi delle modifiche più significative presenti nella terza edizione:

- Abbiamo aggiunto i nuovi capitoli sugli alberi di van Emde Boas e sugli algoritmi multithread; abbiamo dedicato una nuova appendice ai concetti di base delle matrici.
- Abbiamo rivisto il capitolo sulle ricorrenze per trattare in modo più generale la tecnica divide et impera; i primi paragrafi spiegano come applicare questa tecnica per risolvere due problemi. Un altro paragrafo di questo capitolo presenta l'algoritmo di Strassen per il prodotto di matrici, che prima era trattato nel capitolo dedicato alle operazioni con le matrici.

- Abbiamo tolto due capitoli che erano raramente trattati nei corsi: gli heap binomiali e le reti di ordinamento. Un concetto chiave del capitolo sulle reti di ordinamento, il principio 0-1, appare nel Problema 8-7 della terza edizione come lemma dell'ordinamento 0-1 per gli algoritmi confronta-scambia. L'analisi degli heap di Fibonacci non si basa più sugli heap binomiali.
- Abbiamo rivisto la nostra trattazione della programmazione dinamica e gli algoritmi golosi. Adesso la programmazione dinamica inizia con un problema più interessante, il taglio delle aste, anziché con la programmazione delle catene di montaggio della seconda edizione. Inoltre, abbiamo trattato un po' più approfonditamente la tecnica dell'annotazione rispetto alla seconda edizione e abbiamo presentato il concetto di grafo dei sottoproblemi come un modo per capire il tempo di esecuzione di un algoritmo di programmazione dinamica. Nel primo esempio di algoritmi golosi, il problema della programmazione delle attività, abbiamo affrontato in modo più diretto la strategia golosa rispetto alla seconda edizione.
- Il modo in cui viene cancellato un nodo da un albero binario di ricerca (che include gli alberi rosso-neri) adesso garantisce che il nodo sia effettivamente cancellato. Nelle prime due edizioni, in alcuni casi, qualche altro nodo veniva cancellato e i suoi contenuti venivano trasferiti al nodo passato alla procedura di cancellazione. Con il nuovo modo di cancellare i nodi nell'albero, se altre componenti di un programma gestiscono i puntatori ai nodi dell'albero, non potrà accadere che esse si trovino erroneamente ad avere puntatori "obsoleti" a nodi che sono stati cancellati.
- La trattazione delle reti di flusso adesso si basa interamente sugli archi. Questo approccio è più intuitivo di quello utilizzato nelle precedenti edizioni del libro.
- Avendo dedicato una nuova appendice alle matrici e avendo trattato l'algoritmo di Strassen in un altro capitolo, il capitolo sulle operazioni con le matrici adesso è più piccolo rispetto alla seconda edizione.
- Abbiamo modificato la trattazione dell'algoritmo di Knuth-Morris-Pratt per le operazioni di string-matching.
- Abbiamo corretto vari errori, molti dei quali ci sono stati segnalati nel nostro sito web della seconda edizione.
- Per soddisfare numerose richieste, abbiamo cambiato la sintassi del nostro pseudocodice. Adesso utilizziamo i segni "=" per indicare l'assegnazione e "==" per il test di uguaglianza, come nei linguaggi C, C++, Java e Python. Abbiamo anche eliminato le parole chiave **do** e **then** e abbiamo utilizzato "//" come simbolo di commento delle righe dello pseudocodice. Abbiamo adottato il punto (.) per indicare gli attributi degli oggetti. Il nostro pseudocodice resta procedurale, anziché orientato agli oggetti. In altre parole, anziché eseguire i metodi sugli oggetti, chiamiamo semplicemente le procedure passando gli oggetti come parametri.
- Abbiamo aggiunto 100 esercizi e 28 problemi nuovi. Abbiamo aggiornato molti riferimenti bibliografici e ne abbiamo aggiunti di nuovi.
- Infine, abbiamo rivisto l'intero libro, modificando frasi e interi paragrafi per rendere più chiaro e attivo lo stile di scrittura.

## Sito web

Potete utilizzare il nostro sito web, <http://mitpress.mit.edu/algorithms/>, per ottenere informazioni supplementari e per comunicare con noi. Il sito web contiene dei link per una lista di errori, per le soluzioni di esercizi e problemi e per altri argomenti. Il sito spiega anche come segnalare errori o suggerimenti.

## Ringraziamenti

Sono più di vent'anni ormai che lavoriamo per MIT Press, ed è stata davvero una fantastica esperienza! Ringraziamo Ellen Faran, Bob Prior, Ada Brunstein e Mary Reilly per il loro aiuto e supporto.

Eravamo geograficamente lontani mentre lavoravamo alla terza edizione, trovandoci nel Dartmouth College Department of Computer Science, nel MIT Computer Science and Artificial Intelligence Laboratory e nel Columbia University Department of Industrial Engineering and Operations Research. Ringraziamo le nostre rispettive università e i nostri colleghi per il supporto e gli stimoli che ci hanno fornito.

Julie Sussman, P.P.A., ancora una volta ci ha tolto l'impaccio della redazione tecnica. Siamo rimasti sorpresi dal numero di errori che ci sono sfuggiti, ma che sono stati catturati da Julie. Il suo lavoro ci è servito per migliorare anche la nostra esposizione. Se ci fosse una Hall of Fame per i redattori tecnici, Julie sarebbe certamente eletta alla prima votazione. È stata davvero fenomenale. Grazie, grazie, grazie Julie! Anche Priya Natarajan ha trovato alcuni errori che abbiamo potuto eliminare prima che il libro andasse in stampa. Gli eventuali errori che sono rimasti sono da attribuire agli autori (e probabilmente sono stati commessi dopo che Julie ha letto le bozze).

La trattazione degli alberi di van Emde Boas è tratta dalle note di Erik Demaine che, a sua volta, si è ispirato a Michael Bender. In questa edizione abbiamo utilizzato anche alcuni concetti tratti dalle opere di Javed Aslam, Bradley Kuszmaul e Hui Zha. Il capitolo sul multithreading si basa sulle note originariamente scritte insieme con Harald Prokop. I testi si rifanno al lavoro di molti altri autori che partecipano al progetto Cilk del MIT, tra i quali Bradley Kuszmaul e Matteo Frigo. La realizzazione dello pseudocodice multithread ha subito l'influenza del progetto Cilk del MIT e delle estensioni Cilk++ al C++ distribuite da Cilk Arts.

Ringraziamo anche i molti lettori della prima e della seconda edizione che hanno segnalato gli errori o dato suggerimenti per migliorare questo libro. Abbiamo corretto tutti gli errori che ci sono stati segnalati e abbiamo accolto molti suggerimenti. Siamo contenti perché il numero di questi lettori è aumentato notevolmente, al punto però che è impossibile elencarli tutti.

Infine, ringraziamo le nostre mogli – Nicole Cormen, Wendy Leiserson, Gail Rivest e Rebecca Ivry – e i nostri figli – Ricky, Will, Debby e Katie Leiserson; Alex e Christopher Rivest; Molly, Noah e Benjamin Stein – per l'amore e il supporto che ci hanno dato durante la scrittura di questo libro. La pazienza e l'incoraggiamento dei nostri familiari hanno reso possibile questo progetto. A loro dedichiamo con affetto questo libro.

THOMAS H. CORMEN  
CHARLES E. LEISERSON  
RONALD L. RIVEST  
CLIFFORD STEIN

*Lebanon, New Hampshire  
Cambridge, Massachusetts  
Cambridge, Massachusetts  
New York, New York*

Febbraio 2009

## Presentazione dell'edizione italiana

Cosa può aggiungere il curatore della traduzione di un libro di così grande successo che non sia già stato ampiamente detto nella prefazione degli autori? Mi limiterò a esporre le ragioni per cui ritengo importante una traduzione di tale libro in italiano e le ragioni di alcune scelte adottate nella traduzione.

Nella mia lunga esperienza didattica nell'insegnamento di corsi di Algoritmi ho avuto occasione di visionare molti testi, alcuni scritti direttamente in italiano, altri tradotti dall'inglese, altri ancora nella loro versione originale in inglese. La maggior parte di essi appartiene a due categorie.

Una prima categoria è costituita dai cosiddetti "testi sacri". I testi in questa prima categoria sono troppo "difficili" per essere usati come libro di testo in un corso universitario di primo livello. Essi sono più adatti come testi di riferimento per studiosi della materia e per professionisti di alto livello. Una seconda categoria è costituita da testi che trattano gli algoritmi nel contesto dell'insegnamento di uno specifico linguaggio di programmazione prescelto all'interno del corso universitario.

Il libro di Cormen, Leiserson, Rivest e Stein si colloca nel mezzo tra queste due categorie: rispetto ai testi sacri la trattazione della materia è stata semplificata omettendo quegli argomenti e approfondimenti la cui trattazione richiede delle nozioni matematiche di più alto livello, pur mantenendo la stessa rigosità di trattazione degli argomenti non omessi. Questo lo rende accessibile a qualsiasi studente di un corso di laurea scientifico di primo livello interessato alla materia, e certamente a ogni studente di un corso di laurea in Informatica.

Rispetto ai testi della seconda categoria la materia trattata è molto più ampia e questo ha diversi vantaggi. Al docente di un corso di algoritmi permette un'ampia possibilità di scelta degli argomenti da trattare a lezione consentendogli di adattare il corso sia al tipo di uditorio sia alle sue preferenze personali. Per gli studenti particolarmente interessati alla materia (forse pochi) costituisce un utile strumento per allargare le conoscenze. Per gli studenti di Informatica che diventeranno informatici professionisti (sperabilmente molti) esso costituirà un utile riferimento per tutti quei problemi algoritmici che inevitabilmente si presenteranno loro nello svolgimento della professione.

Nel corso del lavoro di traduzione e di curatela si sono dovute fare alcune scelte di carattere generale. La prima scelta sofferta è stata quella di trovare un giusto compromesso tra una traduzione che rimanesse più aderente possibile al testo originale e una traduzione libera che mantenesse soltanto il senso del testo tradotto. Se si fosse trattato di un testo letterario la scelta sarebbe stata senz'altro la seconda in quanto lo stile della prosa italiana sarebbe stato di primaria importanza. Trattandosi di un testo tecnico si è ritenuto che l'aderenza al testo originale fosse importante anche se questo avrebbe inevitabilmente appesantito la prosa italiana con qualche reminiscenza delle costruzioni sintattiche inglesi. Crediamo di aver raggiunto un buon compromesso.

Un'altra decisione sofferta è stata quella relativa alla traduzione dei programmi scritti in pseudocodice. Nessun dubbio sul fatto che le parole chiave, **while**, **if**, **else** ecc., non dovessero essere tradotte. Nessun dubbio neppure sul fatto che le istruzioni astratte e i commenti inseriti nello pseudocodice andassero invece tradotti in italiano.

Il problema era se tradurre anche gli identificatori (nomi dei programmi, delle variabili ecc.) usati nello pseudocodice. Da una parte tradurre il nome SORT di un programma che ordina un array con ORDINA avrebbe certamente facilitato la comprensione da parte di studenti completamente digiuni di inglese (sperabilmente una minoranza). Dall'altra, qualunque implementazione di tale algoritmo presente in una libreria di programmi si chiamerà certamente SORT e non ORDINA. Tradurre tali identificatori avrebbe quindi soltanto spostato (e non eliminato) la difficoltà di tali studenti al momento successivo in cui cercare una implementazione dell'algoritmo.

Abbiamo quindi deciso di lasciare nella loro forma originale inglese gli identificatori. Naturalmente un bravo docente, all'occorrenza, non mancherà di spiegare il significato dei termini inglesi usati come identificatori.

*Livio Colussi \**  
*Padova, marzo 2010*

---

\* **Livio Colussi** è professore associato di Informatica presso la Facoltà di Scienze MM.FF.NN. dell'Università degli Studi di Padova, dove insegna alcuni corsi di Algoritmi a studenti dei corsi di Laurea triennale e specialistica in Informatica.