

# Indice generale

<b>Introduzione .....</b>	<b>xi</b>
A chi è rivolto questo libro? .....	xi
Una nota sul codice .....	xii
Contenuti trattati .....	xii
Come leggere questo libro .....	xiii
Ringraziamenti .....	xv
<b>Capitolo 1 Perché le strutture dati sono così importanti .....</b>	<b>1</b>
Strutture dati .....	2
L'array: la struttura dati fondamentale .....	3
Operazioni su una struttura dati .....	4
Misurazione della velocità .....	4
Lettura .....	5
Ricerca .....	8
Inserimento .....	10
Eliminazione .....	13
Il set: come una singola regola può influire sull'efficienza .....	14
Conclusioni .....	17
Esercizi .....	18
<b>Capitolo 2 Perché gli algoritmi sono così importanti .....</b>	<b>19</b>
Array ordinati .....	20
Ricerca in un array ordinato .....	22
Ricerca binaria .....	24
Implementazione della ricerca binaria .....	26
Ricerca binaria vs. ricerca lineare .....	28
Un piccolo quiz .....	29
Conclusioni .....	30
Esercizi .....	30

<b>Capitolo 3</b>	<b>La notazione Big O.....</b>	<b>31</b>
	Big O: quanti passaggi per N elementi?.....	32
	Il cuore di Big O.....	33
	Più in profondità nel cuore di Big O.....	34
	Stesso algoritmo, scenari differenti.....	35
	Un algoritmo del terzo tipo.....	35
	I logaritmi.....	37
	Parliamo di $O(\log N)$ .....	37
	Esempi pratici.....	38
	Conclusioni.....	39
	Esercizi.....	40
<b>Capitolo 4</b>	<b>Velocizzare il codice grazie al Big O .....</b>	<b>43</b>
	L'algoritmo Bubble Sort.....	43
	Il Bubble Sort in azione.....	44
	Implementazione del Bubble Sort.....	48
	L'efficienza del Bubble Sort.....	50
	Un problema quadratico.....	52
	Una soluzione lineare.....	53
	Conclusioni.....	55
	Esercizi.....	55
<b>Capitolo 5</b>	<b>Ottimizzare il codice con e senza Big O.....</b>	<b>57</b>
	L'algoritmo Selection Sort.....	57
	Il Selection Sort in azione.....	58
	Implementazione del Selection Sort.....	63
	L'efficienza del Selection Sort.....	65
	Ignorare le costanti.....	65
	Categorie Big O.....	66
	Un esempio pratico.....	68
	Passi significativi.....	69
	Conclusioni.....	69
	Esercizi.....	70
<b>Capitolo 6</b>	<b>Ottimizzare per scenari meno pessimisti.....</b>	<b>71</b>
	L'algoritmo Insertion Sort.....	71
	L'Insertion Sort in azione.....	73
	Implementazione dell'Insertion Sort.....	77
	L'efficienza dell'Insertion Sort.....	79
	Il caso medio.....	81
	Un esempio pratico.....	83
	Conclusioni.....	85
	Esercizi.....	85

<b>Capitolo 7</b>	<b>La notazione Big O nel codice di tutti i giorni.....87</b>	
	Media di numeri pari.....	88
	Un costruttore di parole.....	89
	Estrazioni da un array.....	91
	Calcolo della media delle temperature.....	91
	Etichette per vestiti.....	92
	Conteggio degli “1”.....	93
	Controllo dei palindromi.....	94
	Calcolo di tutti i prodotti.....	94
	Gestire più dataset.....	96
	Cracker di password.....	97
	Conclusioni.....	99
	Esercizi.....	99
<b>Capitolo 8</b>	<b>Accelerare (molto) le ricerche con le tabelle hash...103</b>	
	Le tabelle hash.....	104
	Funzioni hash.....	104
	Costruire un dizionario dei sinonimi per divertimento e profitto (ma soprattutto profitto).....	105
	Ricerche in una tabella hash.....	107
	Ricerche unidirezionali.....	108
	Affrontare le collisioni.....	108
	Creare una tabella hash efficiente.....	111
	Un grande atto di equilibrio.....	112
	Tabelle hash per l’organizzazione.....	113
	Tabelle hash per la velocità.....	114
	Sottoinsiemi di array.....	115
	Conclusioni.....	118
	Esercizi.....	119
<b>Capitolo 9</b>	<b>Codice più elegante grazie agli stack e alle code ....121</b>	
	Gli stack.....	122
	Tipi di dati astratti.....	124
	Uno stack in azione.....	125
	Implementazione: un linter di codice basato su uno stack.....	128
	L’importanza delle strutture dati vincolate.....	130
	Riepilogo sugli stack.....	131
	Le code.....	131
	Implementazione della coda.....	132
	Una coda in azione.....	133
	Conclusioni.....	134
	Esercizi.....	134
<b>Capitolo 10</b>	<b>Ricorrere ricorsivamente con la ricorsione .....135</b>	
	Ricorsione al posto dei cicli.....	135

Il caso base.....	136
Lettura del codice ricorsivo.....	137
La ricorsione vista con gli occhi del computer.....	139
Lo stack delle chiamate.....	140
Overflow dello stack.....	142
Attraversamento del file system.....	142
Conclusioni.....	143
Esercizi.....	144

## **Capitolo 11 Imparare a scrivere codice ricorsivo.....147**

Problemi ricorsivi: ripetizione delle esecuzioni.....	147
Il trucco ricorsivo: passaggio di argomenti.....	148
Categoria ricorsiva: calcoli.....	151
Due approcci ai calcoli.....	152
Ricorsione top-down: un nuovo modo di pensare.....	153
Il processo di pensiero top-down.....	154
Somma di un array.....	154
Inversione di stringhe.....	156
Contare le “X”.....	157
Il problema delle scale.....	158
Il caso base del problema delle scale.....	160
Generazione di anagrammi.....	162
L’efficienza della generazione di anagrammi.....	164
Conclusioni.....	165
Esercizi.....	165

## **Capitolo 12 Programmazione dinamica.....167**

Chiamate ricorsive non necessarie.....	167
Esame della funzione max() ricorsiva.....	168
La soluzione è semplice.....	171
L’efficienza della ricorsione.....	171
Sottoproblemi sovrapposti.....	172
Programmazione dinamica attraverso la memoizzazione.....	174
Implementazione della memoizzazione.....	175
Programmazione dinamica bottom-up.....	177
Fibonacci bottom-up.....	177
Memoizzazione vs. bottom-up.....	178
Conclusioni.....	179
Esercizi.....	179

## **Capitolo 13 Algoritmi ricorsivi per migliorare la velocità.....181**

Il partizionamento.....	182
Implementazione del partizionamento.....	184
L’algoritmo Quicksort.....	186
Implementazione del Quicksort.....	191

L'efficienza del Quicksort.....	192
Il Quicksort visto dall'alto.....	193
L'efficienza Big O del Quicksort.....	194
Il Quicksort nello scenario peggiore .....	196
Quicksort vs. Insertion Sort .....	197
L'algoritmo Quickselect.....	198
L'efficienza del Quickselect.....	199
Implementazione del Quickselect .....	200
L'ordinamento e gli altri algoritmi.....	200
Conclusioni .....	202
Esercizi .....	202

## **Capitolo 14 Strutture dati basate su nodi .....203**

Liste concatenate .....	203
Implementazione di una lista concatenata.....	205
La lettura in una lista concatenata.....	207
Implementazione della lettura di liste concatenate .....	207
La ricerca in una lista concatenata .....	208
Implementazione della ricerca nelle liste concatenate .....	209
L'inserimento in una lista concatenata .....	209
Implementazione dell'inserimento nelle liste concatenate .....	212
L'eliminazione in una lista concatenata .....	213
Implementazione dell'eliminazione nelle liste concatenate.....	214
L'efficienza delle operazioni sulle liste concatenate.....	215
Le liste concatenate in azione.....	216
Liste a doppio concatenamento .....	217
Implementazione dell'inserimento nelle liste a doppio concatenamento .....	218
Andare avanti e indietro.....	219
Le code come liste a doppio concatenamento.....	219
Implementazione della coda con una lista a doppio concatenamento .....	219
Conclusioni .....	221
Esercizi .....	221

## **Capitolo 15 Accelerare l'esecuzione con gli alberi binari.....223**

Gli alberi .....	224
Alberi binari di ricerca.....	226
Ricerca in un albero binario di ricerca .....	227
L'efficienza della ricerca in un albero binario di ricerca.....	229
Log(N) livelli.....	229
Implementazione della ricerca in un albero binario di ricerca .....	230
Inserimento in un albero binario di ricerca.....	231
Implementazione dell'inserimento in un albero binario	

di ricerca .....	233
L'ordine di inserimento.....	234
Eliminazione in un albero binario di ricerca.....	235
Eliminazione di un nodo con due figli.....	237
Trovare il nodo successore.....	237
Nodo successore con un figlio destro .....	239
Il nodo successore è un figlio destro .....	240
L'algoritmo di eliminazione per intero .....	240
Implementazione dell'eliminazione di un nodo dall'albero binario di ricerca.....	241
L'efficienza dell'eliminazione dell'albero binario di ricerca.....	245
Alberi binari di ricerca in azione.....	245
Attraversamento dell'albero di ricerca binaria .....	246
Conclusioni .....	249
Esercizi .....	249

## **Capitolo 16 Mantenere le priorità con gli heap .....251**

Code di priorità.....	251
Gli heap.....	253
La condizione dello heap .....	253
Alberi completi .....	254
Proprietà dello heap.....	256
Inserimento nello heap .....	257
Alla ricerca dell'ultimo nodo.....	259
Eliminazione di un nodo dallo heap.....	260
Heap vs. array ordinati .....	264
Di nuovo sul problema dell'ultimo nodo .....	265
Array come heap .....	267
Attraversamento di uno heap basato su array .....	269
Implementazione dell'inserimento nello heap .....	270
Implementazione dell'eliminazione dallo heap .....	271
Implementazioni alternative di uno heap.....	273
Heap come code di priorità .....	273
Conclusioni .....	273
Esercizi .....	274

## **Capitolo 17 Ricercare testi con i trie .....275**

I trie.....	276
Il nodo del trie .....	276
La classe Trie.....	277
Memorizzazione di parole.....	277
La necessità dell'asterisco .....	279
Ricerche in un trie .....	281
Implementazione della ricerca nel trie.....	284

L'efficienza della ricerca nel trie .....	284
Inserimenti in un trie.....	285
Implementazione dell'inserimento di Trie .....	289
Realizzazione del completamento automatico .....	289
Raccogliere tutte le parole .....	290
Descrizione dettagliata della ricorsione .....	291
Completamento del completamento automatico .....	295
Trie con valori: un completamento automatico migliore.....	296
Conclusioni .....	296
Esercizi .....	297
<b>Capitolo 18 Collegare tutto con i grafi .....</b>	<b>299</b>
I grafi .....	300
Grafì e alberi .....	300
Il gergo dei grafi .....	301
Implementazione semplificata di un grafo .....	301
Grafì diretti.....	302
Implementazione a oggetti dei grafi .....	302
Ricerca nei grafi .....	304
Ricerca in profondità (depth-first).....	306
Panoramica della ricerca in profondità.....	307
Implementazione della ricerca in profondità.....	314
Ricerca in ampiezza (breadth-first).....	315
Panoramica della ricerca in ampiezza.....	316
Implementazione della ricerca in ampiezza.....	324
Ricerca in profondità o in ampiezza? .....	326
L'efficienza della ricerca nei grafi.....	327
$O(V + E)$ .....	328
Grafì ponderati .....	330
Codice per i grafì ponderati .....	331
Il problema del percorso piú breve .....	332
L'algoritmo di Dijkstra .....	333
Impostazione dell'algoritmo di Dijkstra.....	333
I passaggi dell'algoritmo di Dijkstra.....	334
Esempio d'uso dell'algoritmo di Dijkstra.....	334
Trovare il percorso piú breve.....	341
Implementazione dell'algoritmo di Dijkstra.....	342
L'efficienza dell'algoritmo di Dijkstra .....	346
Conclusioni .....	347
Esercizi .....	347
<b>Capitolo 19 Gestire i vincoli di spazio.....</b>	<b>351</b>
Il Big O della complessità spaziale .....	351

Compromessi tra tempo e spazio.....	354
I costi nascosti della ricorsione .....	356
Conclusioni .....	358
Esercizi .....	358

**Capitolo 20 Tecniche per l'ottimizzazione del codice .....361**

Prerequisito: determinate il vostro attuale Big O.....	361
Partite da qui: il miglior Big O immaginabile .....	362
Far lavorare l'immaginazione .....	363
Ricerche magiche.....	363
Cercare magicamente gli autori .....	363
Una nuova struttura dati .....	365
Il problema della somma di due numeri .....	366
Individuazione degli schemi.....	369
Il gioco delle monete.....	369
Generazione di esempi.....	370
Il problema degli scambi e delle somme .....	371
Algoritmi "avidì" .....	375
Il massimo di un array.....	376
La sottosezione maggiore.....	377
Previsioni di azioni di borsa e avidità.....	382
Cambiare la struttura dati.....	387
Il controllo degli anagrammi.....	387
Raggruppamenti .....	390
Conclusioni .....	392
Esercizi .....	392

**Indice analitico.....395**