

Prefazione		xv
1	Algoritmi e loro implementazione in Java	1
1.1	Ciclo di sviluppo di codice algoritmico	2
1.2	Fase progettuale	3
1.2.1	Analisi dei requisiti	4
1.2.2	Valutazione della difficoltà intrinseca del problema	4
1.2.3	Progetto di un algoritmo risolutivo	4
1.2.4	Analisi di correttezza	5
1.2.5	Analisi delle prestazioni in un modello di costo teorico	5
1.2.6	Progetto di algoritmi più efficienti	9
1.3	Fase realizzativa	10
1.3.1	Implementazione in un linguaggio di programmazione	10
1.3.2	Collaudo e analisi sperimentale	12
1.3.3	Metodologie di analisi sperimentale	15
1.3.4	Messa a punto e ingegnerizzazione	19
1.4	Ingredienti per l'implementazione in Java	21
1.4.1	Stessa interfaccia, diverse realizzazioni	21
1.4.2	Tipi generici e boxing	23
1.4.3	Test di uguaglianza: il metodo equals	24
1.4.4	Ordinamento naturale: il metodo compareTo	25
1.4.5	Gestione degli errori: eccezioni	27
1.4.6	Iteratori	29
1.4.7	Il Java Collections Framework	30
1.5	Progetto motore di ricerca	31
1.6	Sommario	34
1.7	Esercizi	36
2	Strutture dati elementari	37
2.1	Tipi di dato e interfacce Java	37
2.2	Strutture dati e classi Java	38
2.3	Tecniche per rappresentare collezioni di oggetti	43
2.3.1	Strutture indicizzate: array	43
2.3.2	Strutture collegate: record e puntatori	47
2.4	Tipo di dato pila	52
2.5	Tipo di dato coda	55

2.6	Progetto motore di ricerca: lexicon e indice diretto	58
2.7	Analisi sperimentale	63
2.8	Sommario	68
2.9	Esercizi	69
3	Alberi	71
3.1	Definizioni preliminari sugli alberi	71
3.2	Tipo di dato albero	72
3.3	Tipo di dato albero binario	75
3.4	Rappresentazioni indicizzate	77
3.4.1	Vettore padri	78
3.4.2	Vettore posizionale	81
3.5	Rappresentazioni collegate	82
3.5.1	Puntatori ai figli	83
3.5.2	Lista figli	86
3.5.3	Primo figlio-fratello successivo	86
3.6	Visite di alberi	88
3.6.1	Visita in profondità	89
3.6.2	Visita in ampiezza	92
3.7	Progetto motore di ricerca: compressione	94
3.7.1	I codici di Huffman	94
3.7.2	Gli alberi di Huffman	95
3.8	Analisi sperimentale	100
3.9	Sommario	102
3.10	Esercizi	103
4	Ordinamento	105
4.1	Ordinare in tempo quadratico	106
4.1.1	Ordinamenti incrementali	106
4.1.2	Ordinamento a bolle	109
4.2	Mergesort	110
4.3	Heapsort	113
4.3.1	Struttura dati heap	113
4.3.2	Ordinare <i>in loco</i> mediante heap	117
4.4	Quicksort	118
4.5	Ordinare in tempo lineare	121
4.5.1	Integersort	122
4.5.2	Radixsort	123
4.6	Progetto motore di ricerca: indice inverso	126
4.7	Analisi sperimentale	132
4.8	Sommario	136
4.9	Esercizi	137

5	Selezione	141
5.1	Selezione per piccoli valori di k	141
5.1.1	Ricerca del secondo minimo	142
5.1.2	L'algoritmo Heapsselect	143
5.2	Calcolo randomizzato del mediano	144
5.3	Calcolo deterministico del mediano	146
5.4	Progetto motore di ricerca: ranking	151
5.5	Analisi sperimentale	157
5.6	Sommario	160
5.7	Esercizi	160
6	Alberi di ricerca	165
6.1	Alberi binari di ricerca	167
6.2	Alberi AVL	177
6.2.1	Ribilanciamento tramite rotazioni	178
6.2.2	Realizzazione di rotazioni semplici e doppie	181
6.2.3	Inserimenti e cancellazioni di chiavi	185
6.3	Alberi 2-3	188
6.4	B-alberi	193
6.5	Alberi 2-3-4 e alberi rosso-neri	198
6.6	Analisi sperimentale	205
6.7	Sommario	212
6.8	Esercizi	212
7	Tabelle hash	217
7.1	Tabelle ad accesso diretto	217
7.2	Tabelle hash	220
7.2.1	Hashing perfetto	221
7.2.2	Altre tecniche di hashing	223
7.3	Liste di collisione	227
7.4	Indirizzamento aperto	231
7.4.1	Funzioni di scansione	235
7.4.2	Generazione di numeri primi	239
7.4.3	Cancellazione di chiavi	240
7.4.4	Analisi del costo di scansione	243
7.5	Analisi sperimentale	244
7.6	Sommario	251
7.7	Esercizi	253
8	Code con priorità	255
8.1	d -heap	255
8.2	Heap binomiali	268
8.3	Analisi sperimentale	279
8.4	Sommario	286
8.5	Esercizi	287

9	Union-find	289
9.1	Approcci elementari al problema union-find	294
9.1.1	Algoritmi di tipo QuickFind	294
9.1.2	Algoritmi di tipo QuickUnion	296
9.2	Bilanciamento nell'operazione union	298
9.2.1	Bilanciamento per algoritmi di tipo QuickFind	299
9.2.2	Bilanciamento per algoritmi di tipo QuickUnion	301
9.3	Euristiche di compressione nell'operazione find	306
9.4	Analisi sperimentale	309
9.5	Sommario	315
9.6	Esercizi	316
10	Grafi e visite di grafi	319
10.1	Definizioni preliminari sui grafi	321
10.2	Tipo di dato grafo	323
10.3	Strutture dati per rappresentare grafi	327
10.3.1	Lista di archi	327
10.3.2	Liste di adiacenza	329
10.3.3	Matrice di adiacenza	336
10.3.4	Matrice di incidenza	337
10.4	Visite di grafi	338
10.5	Visita in ampiezza	344
10.6	Visita in profondità	351
10.7	Progetto motore di ricerca: crawling	359
10.8	Analisi sperimentale	364
10.9	Sommario	367
10.10	Esercizi	368
11	Minimo albero ricoprente	371
11.1	Proprietà dei minimi alberi ricoprenti	372
11.2	Algoritmo di Kruskal	374
11.2.1	Implementazione mediante union-find	375
11.3	Algoritmo di Prim	379
11.3.1	Implementazione mediante code con priorità	382
11.4	Progetto motore di ricerca: clustering	386
11.5	Sommario	391
11.6	Esercizi	392
12	Cammini minimi	395
12.1	Cammini minimi e distanze in un grafo	395
12.2	La tecnica del rilassamento	398
12.3	Algoritmo di Bellman, Ford e Moore	399
12.3.1	Implementazione mediante coda	400
12.4	Algoritmo di Dijkstra	404
12.4.1	Implementazione mediante code con priorità	407
12.5	Algoritmo di Floyd e Warshall	412

12.6 Progetto motore di ricerca: crawling	414
12.7 Analisi sperimentale	416
12.8 Sommario	418
12.9 Esercizi	419
Bibliografia	421