Algorithm Design:
Foundations, Analysis, and Internet Examples

Michael T. Goodrich
Department of Information and Computer Science
University of California, Irvine

Roberto Tamassia
Department of Computer Science
Brown University
Contents

1 Fundamental Tools 1

1 Algorithm Analysis 3
  1.1 Methodologies for Analyzing Algorithms 5
  1.2 Asymptotic Notation 13
  1.3 A Quick Mathematical Review 21
  1.4 Case Studies in Algorithm Analysis 31
  1.5 Amortization 34
  1.6 Experimentation 42
  1.7 Exercises 47

2 Basic Data Structures 55
  2.1 Stacks and Queues 57
  2.2 Vectors, Lists, and Sequences 65
  2.3 Trees 75
  2.4 Priority Queues and Heaps 94
  2.5 Dictionaries and Hash Tables 114
  2.6 Java Example: Heap 128
  2.7 Exercises 131

3 Search Trees and Skip Lists 139
  3.1 Ordered Dictionaries and Binary Search Trees 141
  3.2 AVL Trees 152
  3.3 Bounded-Depth Search Trees 159
  3.4 Splay Trees 185
  3.5 Skip Lists 195
  3.6 Java Example: AVL and Red-Black Trees 202
  3.7 Exercises 212

4 Sorting, Sets, and Selection 217
  4.1 Merge-Sort 219
  4.2 The Set Abstract Data Type 225
  4.3 Quick-Sort 235
  4.4 A Lower Bound on Comparison-Based Sorting 239
  4.5 Bucket-Sort and Radix-Sort 241
  4.6 Comparison of Sorting Algorithms 244
  4.7 Selection 245
  4.8 Java Example: In-Place Quick-Sort 248
  4.9 Exercises 251
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Fundamental Techniques</td>
<td>257</td>
</tr>
<tr>
<td>5.1</td>
<td>The Greedy Method</td>
<td>259</td>
</tr>
<tr>
<td>5.2</td>
<td>Divide-and-Conquer</td>
<td>263</td>
</tr>
<tr>
<td>5.3</td>
<td>Dynamic Programming</td>
<td>274</td>
</tr>
<tr>
<td>5.4</td>
<td>Exercises</td>
<td>282</td>
</tr>
<tr>
<td>II</td>
<td>Graph Algorithms</td>
<td>285</td>
</tr>
<tr>
<td>6</td>
<td>Graphs</td>
<td>287</td>
</tr>
<tr>
<td>6.1</td>
<td>The Graph Abstract Data Type</td>
<td>289</td>
</tr>
<tr>
<td>6.2</td>
<td>Data Structures for Graphs</td>
<td>296</td>
</tr>
<tr>
<td>6.3</td>
<td>Graph Traversal</td>
<td>303</td>
</tr>
<tr>
<td>6.4</td>
<td>Directed Graphs</td>
<td>316</td>
</tr>
<tr>
<td>6.5</td>
<td>Java Example: Depth-First Search</td>
<td>329</td>
</tr>
<tr>
<td>6.6</td>
<td>Exercises</td>
<td>335</td>
</tr>
<tr>
<td>7</td>
<td>Weighted Graphs</td>
<td>339</td>
</tr>
<tr>
<td>7.1</td>
<td>Single-Source Shortest Paths</td>
<td>341</td>
</tr>
<tr>
<td>7.2</td>
<td>All-Pairs Shortest Paths</td>
<td>354</td>
</tr>
<tr>
<td>7.3</td>
<td>Minimum Spanning Trees</td>
<td>360</td>
</tr>
<tr>
<td>7.4</td>
<td>Java Example: Dijkstra's Algorithm</td>
<td>373</td>
</tr>
<tr>
<td>7.5</td>
<td>Exercises</td>
<td>376</td>
</tr>
<tr>
<td>8</td>
<td>Network Flow and Matching</td>
<td>381</td>
</tr>
<tr>
<td>8.1</td>
<td>Flows and Cuts</td>
<td>383</td>
</tr>
<tr>
<td>8.2</td>
<td>Maximum Flow</td>
<td>387</td>
</tr>
<tr>
<td>8.3</td>
<td>Maximum Bipartite Matching</td>
<td>396</td>
</tr>
<tr>
<td>8.4</td>
<td>Minimum-Cost Flow</td>
<td>398</td>
</tr>
<tr>
<td>8.5</td>
<td>Java Example: Minimum-Cost Flow</td>
<td>405</td>
</tr>
<tr>
<td>8.6</td>
<td>Exercises</td>
<td>412</td>
</tr>
<tr>
<td>III</td>
<td>Internet Algorithmics</td>
<td>415</td>
</tr>
<tr>
<td>9</td>
<td>Text Processing</td>
<td>417</td>
</tr>
<tr>
<td>9.1</td>
<td>Strings and Pattern Matching Algorithms</td>
<td>419</td>
</tr>
<tr>
<td>9.2</td>
<td>Tries</td>
<td>429</td>
</tr>
<tr>
<td>9.3</td>
<td>Text Compression</td>
<td>440</td>
</tr>
<tr>
<td>9.4</td>
<td>Text Similarity Testing</td>
<td>443</td>
</tr>
<tr>
<td>9.5</td>
<td>Exercises</td>
<td>447</td>
</tr>
<tr>
<td>10</td>
<td>Number Theory and Cryptography</td>
<td>451</td>
</tr>
<tr>
<td>10.1</td>
<td>Fundamental Algorithms Involving Numbers</td>
<td>453</td>
</tr>
<tr>
<td>10.2</td>
<td>Cryptographic Computations</td>
<td>471</td>
</tr>
<tr>
<td>10.3</td>
<td>Information Security Algorithms and Protocols</td>
<td>481</td>
</tr>
<tr>
<td>10.4</td>
<td>The Fast Fourier Transform</td>
<td>488</td>
</tr>
<tr>
<td>10.5</td>
<td>Java Example: FFT</td>
<td>500</td>
</tr>
<tr>
<td>10.6</td>
<td>Exercises</td>
<td>508</td>
</tr>
</tbody>
</table>
## Contents

### 11 Network Algorithms

11.1 Complexity Measures and Models ............................................... 513
11.2 Fundamental Distributed Algorithms ........................................... 517
11.3 Broadcast and Unicast Routing .................................................... 530
11.4 Multicast Routing .................................................................. 535
11.5 Exercises ............................................................................. 541

### IV Additional Topics

### 12 Computational Geometry

12.1 Range Trees ........................................................................... 549
12.2 Priority Search Trees ............................................................. 556
12.3 Quadtrees and k-D Trees ........................................................ 561
12.4 The Plane Sweep Technique .................................................... 565
12.5 Convex Hulls ......................................................................... 572
12.6 Java Example: Convex Hull ..................................................... 583
12.7 Exercises .............................................................................. 587

### 13 NP-Completeness

13.1 P and NP ............................................................................... 593
13.2 NP-Completeness .................................................................. 599
13.3 Important NP-Complete Problems ........................................... 603
13.4 Approximation Algorithms ...................................................... 618
13.5 Backtracking and Branch-and-Bound ...................................... 627
13.6 Exercises .............................................................................. 638

### 14 Algorithmic Frameworks

14.1 External-Memory Algorithms .................................................. 645
14.2 Parallel Algorithms ................................................................. 657
14.3 Online Algorithms ................................................................ 667
14.4 Exercises .............................................................................. 680

### A Useful Mathematical Facts

### Bibliography

### Index
<table>
<thead>
<tr>
<th>Term</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>δ-approximation</td>
<td>618</td>
</tr>
<tr>
<td>c-incremental</td>
<td>593</td>
</tr>
<tr>
<td>2SAT, 607, 640</td>
<td></td>
</tr>
<tr>
<td>3SAT, 607, 608</td>
<td></td>
</tr>
<tr>
<td><strong>abstract data type</strong></td>
<td></td>
</tr>
<tr>
<td>dictionary, 114–115, 141</td>
<td></td>
</tr>
<tr>
<td>graph, 289–295</td>
<td></td>
</tr>
<tr>
<td>list, 69–70</td>
<td></td>
</tr>
<tr>
<td>partition, 227–234</td>
<td></td>
</tr>
<tr>
<td>priority queue, 94–95, 112–113</td>
<td></td>
</tr>
<tr>
<td>queue, 61</td>
<td></td>
</tr>
<tr>
<td>sequence, 73–74</td>
<td></td>
</tr>
<tr>
<td>set, 225–234</td>
<td></td>
</tr>
<tr>
<td>stack, 57</td>
<td></td>
</tr>
<tr>
<td>string, 419–420</td>
<td></td>
</tr>
<tr>
<td>tree, 77–78</td>
<td></td>
</tr>
<tr>
<td>vector, 65</td>
<td></td>
</tr>
<tr>
<td>(a, b) tree, 650–652</td>
<td></td>
</tr>
<tr>
<td>depth property, 650</td>
<td></td>
</tr>
<tr>
<td>size property, 650</td>
<td></td>
</tr>
<tr>
<td>accepting a string, 594</td>
<td></td>
</tr>
<tr>
<td>access control lists, 472</td>
<td></td>
</tr>
<tr>
<td>accounting method, 36–37</td>
<td></td>
</tr>
<tr>
<td>Achter, 510</td>
<td></td>
</tr>
<tr>
<td>Ackermann function, 234, 256</td>
<td></td>
</tr>
<tr>
<td>acyclic, 316</td>
<td></td>
</tr>
<tr>
<td>additive inverse, 458</td>
<td></td>
</tr>
<tr>
<td>Adel'son-Vel'skii, 216</td>
<td></td>
</tr>
<tr>
<td>adjacency list, 296, 299</td>
<td></td>
</tr>
<tr>
<td>adjacency matrix, 296, 301</td>
<td></td>
</tr>
<tr>
<td>adjacent, 290</td>
<td></td>
</tr>
<tr>
<td>Adleman, 476</td>
<td></td>
</tr>
<tr>
<td>Agarwal, 590</td>
<td></td>
</tr>
<tr>
<td>Aggarwal, 683</td>
<td></td>
</tr>
<tr>
<td>Aho, 137, 216, 256, 450, 510, 642</td>
<td></td>
</tr>
<tr>
<td>Ahuja, 338, 379, 414</td>
<td></td>
</tr>
<tr>
<td>Ajtai, 683</td>
<td></td>
</tr>
<tr>
<td>Akl, 683</td>
<td></td>
</tr>
<tr>
<td>algorithm, 4</td>
<td></td>
</tr>
<tr>
<td>algorithm analysis, 8–33</td>
<td></td>
</tr>
<tr>
<td>average case, 11</td>
<td></td>
</tr>
<tr>
<td>worst case, 11</td>
<td></td>
</tr>
<tr>
<td>alphabet, 420</td>
<td></td>
</tr>
<tr>
<td>accounting method, 36–37</td>
<td></td>
</tr>
<tr>
<td>potential function, 37–38</td>
<td></td>
</tr>
<tr>
<td>ancestor, 75, 315</td>
<td></td>
</tr>
<tr>
<td>anchor point, 578–581</td>
<td></td>
</tr>
<tr>
<td>antipodal, 665</td>
<td></td>
</tr>
<tr>
<td>antisymmetric property, 94</td>
<td></td>
</tr>
<tr>
<td>anycast, 544</td>
<td></td>
</tr>
<tr>
<td>approximation algorithm, 618</td>
<td></td>
</tr>
<tr>
<td>approximation algorithms, 618–626</td>
<td></td>
</tr>
<tr>
<td>Aragon, 590</td>
<td></td>
</tr>
<tr>
<td>arc, 289</td>
<td></td>
</tr>
<tr>
<td>Archimedes, 4, 54</td>
<td></td>
</tr>
<tr>
<td>Ariadne, 288</td>
<td></td>
</tr>
<tr>
<td>Arnold, 137</td>
<td></td>
</tr>
<tr>
<td>array splitting, 663</td>
<td></td>
</tr>
<tr>
<td>art gallery guarding, 283</td>
<td></td>
</tr>
<tr>
<td>Arya, 590</td>
<td></td>
</tr>
<tr>
<td>asymmetric relation, 289</td>
<td></td>
</tr>
<tr>
<td>asymptotic notation, 13–33</td>
<td></td>
</tr>
<tr>
<td>big-Oh, 13–16</td>
<td></td>
</tr>
<tr>
<td>big-Omega, 16</td>
<td></td>
</tr>
<tr>
<td>big-Theta, 16</td>
<td></td>
</tr>
<tr>
<td>little-oh, 18</td>
<td></td>
</tr>
<tr>
<td>little-omega, 18</td>
<td></td>
</tr>
<tr>
<td>Atallah, 683</td>
<td></td>
</tr>
<tr>
<td>auction algorithm, 674–675</td>
<td></td>
</tr>
<tr>
<td>audit trail, 115</td>
<td></td>
</tr>
<tr>
<td>augmenting cycle, 398</td>
<td></td>
</tr>
<tr>
<td>augmenting path, 388</td>
<td></td>
</tr>
<tr>
<td>Aurenhammer, 590</td>
<td></td>
</tr>
<tr>
<td>authenticated dictionary, 482</td>
<td></td>
</tr>
<tr>
<td>AVL tree, 152–158, 206, 569</td>
<td></td>
</tr>
<tr>
<td>balance factor, 206</td>
<td></td>
</tr>
<tr>
<td>height-balence property, 152</td>
<td></td>
</tr>
<tr>
<td>Awerbuch, 544</td>
<td></td>
</tr>
<tr>
<td>Baase, 510</td>
<td></td>
</tr>
<tr>
<td>back edge, 305, 318, 320, 336</td>
<td></td>
</tr>
<tr>
<td>backtracking, 303, 627–631, 642</td>
<td></td>
</tr>
<tr>
<td>Baeza-Yates, 216, 256, 450, 683</td>
<td></td>
</tr>
<tr>
<td>balance factor, 206</td>
<td></td>
</tr>
<tr>
<td>balanced search tree, 162</td>
<td></td>
</tr>
<tr>
<td>Barůvka, 379</td>
<td></td>
</tr>
<tr>
<td>Barůvka's algorithm, 369–372, 528–529</td>
<td></td>
</tr>
<tr>
<td>Baumert, 642</td>
<td></td>
</tr>
<tr>
<td>Bayer, 216, 683</td>
<td></td>
</tr>
<tr>
<td>Bellman, 284, 379</td>
<td></td>
</tr>
<tr>
<td>Bellman–Ford algorithm, 349–351</td>
<td></td>
</tr>
<tr>
<td>Bentley, 137, 284, 590</td>
<td></td>
</tr>
<tr>
<td>Bertrand's Postulate, 126</td>
<td></td>
</tr>
<tr>
<td>best-first search, 632</td>
<td></td>
</tr>
<tr>
<td>BFS, see breadth-first search</td>
<td></td>
</tr>
</tbody>
</table>
BFS tree, 315
biconnected, 307
biconnected component, 307
big integers, 270
big-Oh notation, 13–16, 54
big-Omega notation, 16
big-Theta notation, 16
Binary Euclid’s Algorithm, 457
binary search, 142–145
binary search tree, 145–151
  insertion, 148
  removal, 149–150
  rotation, 155
  trinode restructure, 155
binary tree, 76, 84–92, 220
  complete, 99
  left child, 76
  level, 84
  linked structure, 92
  proper, 76
  right child, 76
  vector representation, 90–91
binomial expansion, 686
bipartite graph, 396
bit commitment, 483
bit vector, 252
blocking, 647
Booch, 137
Boolean circuit, 597
bootstrapping, 161
Borodin, 683
bottleneck, 395
boundary node, 551
Boyer, 54, 450
branch-and-bound, 632–634, 639, 641, 642
Brassard, 54, 642
Bratley, 642
breadth-first search, 313–316, 320, 523–527
Brent, 683
Brent’s theorem, 659, 660
Brigham, 510
broadcast routing, 530–531
brute force, 420
brute-force pattern matching, 420
B-tree, 652–653
bubble-sort, 137
bucket array, 116
bucket-sort, 241–242
Budd, 137
Burger, 683
cache, 645
cache line, 647
caching algorithms, 668–673
call-by-value, 60
capacity rule, 383
Cardelli, 137
Carlsson, 137
Carmichael numbers, 466
Cartesian coordinates, 572
Cartesian tree, 590
Catalan number, 275
certificate, 486, 596
certificate authority, 486
certificate revocation list, 486
character-jump heuristic, 422
Chen, 683
Chernoff bound, 253
child, 75
children, 75
Chomsky normal form, 284
Chvátal, 642
CIRCUIT-SAT, 598, 600–602, 605
Clarkson, 256
clauses, 605
clearable table, 34
CLIQUE, 610, 638
clique, 610
clockwise, 574
closed form, 12
closest pairs, 568–571
closure, 357
CNF, see conjunctive normal form
CNF-SAT, 605–607, 628, 629, 640, 642
co-NP, 595, 597
coefficient form, 488
Cole, 450, 683
collinear, 574
collision resolution, 116, 120–127
collision-resistant, 481
Comer, 544, 683
comparator, 95, 218, 579, 583
radial, 579, 580, 583
competitive search trees, 676–679
complement, 595
complete binary tree, 99
complexity class, 595
component design, 603
composite, 453
compositeness witness function, 466
compression map, 117, 119
computational geometry, 548–590
  closest pairs, 568–571
  convex hull, 572–583
  degeneracies, 575, 583
  orientation, 574–575
  plane sweep, 566–571
proximity, 568
representations, 572–573
segment intersection, 565–568
conditional probability, 29
confidentiality, 452
congruence, 454
congruent, 454
conjunctive normal form, 605, 607
connected components, 292, 306, 315
conservation rule, 383
container, 137
contradiction, 25
contrapositive, 25
convex hull, 572–583, 590
gift wrapping, 578–579
Graham scan algorithm, 580–583
convexity, 573, 576
convolution, 488
Convolution Theorem, 492
Cook, 642
Cook-Levin Theorem, 600
Cooley, 510
coordinates, 549
core memory, 645
Cormen, 216, 338, 379, 414, 510, 642
cost, 398
counterclockwise, 574
Crochemore, 450
cross edge, 315, 318, 320
cryptography, 471–480
El Gamal cryptosystem, 479–480
RSA cryptosystem, 476–478
cursor, 70
cut, 385
cyber-dollar, 36
cycle, 292
directed, 292
cryptograph
public-key, 475–480
DAG, see directed acyclic graph
data integrity, 452, 472
data structure, 4
  secondary, 161
decision problem, 594
decision tree, 239
decorator pattern, 329–332
decrease-and-conquer, see prune-and-search
decryption, 471
degree, 290, 488
Delaunay triangulation, 590
DeMorgan’s Law, 25
Demurjian, 137
depth, 79–80
dependency, 606
dependent, 315
depth-bounded tree, 159
depth-first search, 303–306
descendant, 75, 315
design patterns
  amortization, 39–41
  brute force, 420
  comparator, 95
  decorator, 329–332
dynamic programming, 274–281, 444–446
greedy method, 442
iterator, 74
locator, 112–113
position, 68
prune-and-search, 245–247
template method, 332
destination, 290
DFS, see depth-first search
DFS tree, 305
DFT, see Discrete Fourier Transform
Di Battista, 137, 338
diameter, 135, 523, 665
dictionary, 114–127, 141–216
(2,4) tree, 163–169
abstract data type, 114–115, 141
AVL tree, 152–158
binary search tree, 145–151
hash table, 116–127
log file, 115
lookup table, 142–145
ordered, 114, 141
red-black tree, 170–184
skip-list, 195–202
unordered, 114, 115
update operations, 148, 149, 154, 157, 198, 199
Diffie-Hellman key exchange, 487
digital signature, 471
  El Gamal, 480
  RSA, 477
digraph, 316
Dijkstra, 379
Dijkstra’s algorithm, 342–348, 373–376
directed acyclic graph, 325–327
directed cycle, 316
directed DFS tree, 318
discovery edge, 305, 315, 318, 320
Discrete Fourier Transform, 491–494
discrete logarithm, 479
disjunctive normal form, 606
distance, 341
<table>
<thead>
<tr>
<th>Term</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>distance vector</td>
<td>532</td>
</tr>
<tr>
<td>distance vector algorithm</td>
<td>532-533</td>
</tr>
<tr>
<td>distributed algorithms</td>
<td>514-517</td>
</tr>
<tr>
<td>message-passing model</td>
<td>514-516</td>
</tr>
<tr>
<td>distributed computing</td>
<td></td>
</tr>
<tr>
<td>asynchronous model</td>
<td>515</td>
</tr>
<tr>
<td>synchronous model</td>
<td>515</td>
</tr>
<tr>
<td>divide-and-conquer</td>
<td>219-221, 235, 263-273, 659</td>
</tr>
<tr>
<td>division method</td>
<td>119</td>
</tr>
<tr>
<td>DNF, see disjunctive normal form</td>
<td></td>
</tr>
<tr>
<td>DNF-DISSAT</td>
<td>638</td>
</tr>
<tr>
<td>DNF-SAT</td>
<td>638</td>
</tr>
<tr>
<td>d-node</td>
<td>159</td>
</tr>
<tr>
<td>double black</td>
<td>177</td>
</tr>
<tr>
<td>double hashing</td>
<td>124</td>
</tr>
<tr>
<td>double red</td>
<td>172</td>
</tr>
<tr>
<td>down-heap bubbling</td>
<td>104, 109</td>
</tr>
<tr>
<td>dynamic programming</td>
<td>274-281, 320, 443-446</td>
</tr>
<tr>
<td>Edelsbrunner</td>
<td>590</td>
</tr>
<tr>
<td>edge</td>
<td>289, 573</td>
</tr>
<tr>
<td>destination</td>
<td>290</td>
</tr>
<tr>
<td>end vertices</td>
<td>290</td>
</tr>
<tr>
<td>incident</td>
<td>290</td>
</tr>
<tr>
<td>multiple</td>
<td>291</td>
</tr>
<tr>
<td>origin</td>
<td>290</td>
</tr>
<tr>
<td>outgoing</td>
<td>290</td>
</tr>
<tr>
<td>parallel</td>
<td>291</td>
</tr>
<tr>
<td>self-loop</td>
<td>291</td>
</tr>
<tr>
<td>edge capacity</td>
<td>383</td>
</tr>
<tr>
<td>edge list</td>
<td>296, 298</td>
</tr>
<tr>
<td>edit distance</td>
<td>449</td>
</tr>
<tr>
<td>Edmonds</td>
<td>284, 414</td>
</tr>
<tr>
<td>Edmonds-Karp algorithm</td>
<td>393-395</td>
</tr>
<tr>
<td>Edmonds-Karp augmentations</td>
<td>393</td>
</tr>
<tr>
<td>El-Yaniv</td>
<td>683</td>
</tr>
<tr>
<td>Elliott</td>
<td>510</td>
</tr>
<tr>
<td>El Gamal cryptosystem</td>
<td>479-480</td>
</tr>
<tr>
<td>Emiris</td>
<td>510</td>
</tr>
<tr>
<td>encryption schemes</td>
<td>473</td>
</tr>
<tr>
<td>end vertices</td>
<td>290</td>
</tr>
<tr>
<td>endpoints</td>
<td>290</td>
</tr>
<tr>
<td>energy-balanced trees</td>
<td>676-679</td>
</tr>
<tr>
<td>equivalence class</td>
<td>308</td>
</tr>
<tr>
<td>equivalence relation</td>
<td>308</td>
</tr>
<tr>
<td>Euclid's algorithm</td>
<td>455-457</td>
</tr>
<tr>
<td>Euclid's algorithm binary</td>
<td>457</td>
</tr>
<tr>
<td>Euclid's algorithm extended</td>
<td>464-465</td>
</tr>
<tr>
<td>Euler pseudo-prime</td>
<td>469</td>
</tr>
<tr>
<td>Euler tour</td>
<td>335, 338</td>
</tr>
<tr>
<td>Euler tour traversal</td>
<td>87, 137</td>
</tr>
<tr>
<td>Euler's Theorem</td>
<td>461</td>
</tr>
<tr>
<td>Even</td>
<td>338, 379, 414</td>
</tr>
<tr>
<td>event</td>
<td>28</td>
</tr>
<tr>
<td>expected value</td>
<td>29</td>
</tr>
<tr>
<td>exponent</td>
<td>23</td>
</tr>
<tr>
<td>Extended Euclid's Algorithm</td>
<td>464</td>
</tr>
<tr>
<td>external memory</td>
<td>644-656, 683</td>
</tr>
<tr>
<td>external-memory algorithm</td>
<td>645-656</td>
</tr>
<tr>
<td>external-memory sorting</td>
<td>654-656</td>
</tr>
<tr>
<td>factorial</td>
<td>686</td>
</tr>
<tr>
<td>failure function</td>
<td>425</td>
</tr>
<tr>
<td>fairness</td>
<td>515</td>
</tr>
<tr>
<td>Fast Fourier Transform</td>
<td>488-507, 510</td>
</tr>
<tr>
<td>Fermat's Little Theorem</td>
<td>459</td>
</tr>
<tr>
<td>FFT, see Fast Fourier Transform</td>
<td></td>
</tr>
<tr>
<td>Fibonacci progression</td>
<td>687</td>
</tr>
<tr>
<td>FIFO, 61, 656</td>
<td></td>
</tr>
<tr>
<td>first-in-first-out</td>
<td>61</td>
</tr>
<tr>
<td>Flajolet</td>
<td>54</td>
</tr>
<tr>
<td>flip a random coin</td>
<td>483</td>
</tr>
<tr>
<td>flow, see network flow</td>
<td></td>
</tr>
<tr>
<td>flow network</td>
<td>383</td>
</tr>
<tr>
<td>Floyd, 137</td>
<td></td>
</tr>
<tr>
<td>Floyd-Warshall algorithm</td>
<td>320, 338</td>
</tr>
<tr>
<td>Ford, 379</td>
<td></td>
</tr>
<tr>
<td>Ford-Fulkerson algorithm</td>
<td>387-395</td>
</tr>
<tr>
<td>forest</td>
<td>292</td>
</tr>
<tr>
<td>forward edge</td>
<td>318</td>
</tr>
<tr>
<td>frame</td>
<td>59</td>
</tr>
<tr>
<td>Fulkerson</td>
<td>414</td>
</tr>
<tr>
<td>fully polynomial-time approximation scheme</td>
<td>619</td>
</tr>
<tr>
<td>Fundamental Theorem of Arithmetic</td>
<td>453</td>
</tr>
<tr>
<td>fusion</td>
<td>167, 651, 653</td>
</tr>
<tr>
<td>Gabow, 256</td>
<td></td>
</tr>
<tr>
<td>Galleger</td>
<td>544</td>
</tr>
<tr>
<td>garbage collection</td>
<td>323-324</td>
</tr>
<tr>
<td>mark-sweep</td>
<td>323</td>
</tr>
<tr>
<td>Garey, 642</td>
<td></td>
</tr>
<tr>
<td>Gauss, 22</td>
<td></td>
</tr>
<tr>
<td>Gavril, 642</td>
<td></td>
</tr>
<tr>
<td>GCD, see greatest common divisor generator</td>
<td>462</td>
</tr>
<tr>
<td>generic merge algorithm</td>
<td>226</td>
</tr>
<tr>
<td>geometric sum</td>
<td>687</td>
</tr>
<tr>
<td>Gibbons</td>
<td>338, 379, 414</td>
</tr>
<tr>
<td>gift wrapping</td>
<td>578-579</td>
</tr>
<tr>
<td>Godbole</td>
<td>284</td>
</tr>
<tr>
<td>Golberg</td>
<td>137</td>
</tr>
<tr>
<td>golden ratio</td>
<td>687</td>
</tr>
<tr>
<td>Golomb, 642</td>
<td></td>
</tr>
<tr>
<td>Gonnet</td>
<td>137, 216, 256, 683</td>
</tr>
<tr>
<td>Goodman</td>
<td>590</td>
</tr>
</tbody>
</table>
Goodrich, 590, 683
googol, 17
Gosling, 137
Graham, 54, 379, 590
Graham scan algorithm, 580–583
grammar, 284
graph, 288–338, 340–379
abstract data type, 289–295
acyclic, 316
bipartite, 396
breadth-first search, 313–316, 318–320
connected, 292, 315
data structures, 296–301
adjacency list, 299–301
adjacency matrix, 301
data structures, 296–301
density list, 296–298
dense, 321
depth-first search, 303–306, 318–320
digraph, 316
directed, 289, 290, 316–327
acyclic, 325–327
strongly connected, 316
methods, 293–295
mixed, 290
reachability, 316–317, 320–321
shortest paths, 320–321
simple, 291
traversal, 303–316
undirected, 289, 290
weighted, 340–379
greatest common divisor, 454–457
Euclid's algorithm, 455–457
greedy method, 259–262, 341, 342, 442
greedy-choice, 259, 442
group, 461
guess-and-test, 266–267
Guibas, 216
Guttag, 137
Haken, 284
HAMILTONIAN-CYCLE, 597, 615, 617, 641, 642
Harmonic number, 626
harmonic number, 688
hash code, 117, 118
hash function, 117, 124–126
2-universal, 125
one-way, 481
hash table, 116–127
bucket array, 116
capacity, 116
chaining, 121
clustering, 124
collison, 116
collision resolution, 120–127
double hashing, 124
linear probing, 123
open addressing, 124
quadratic probing, 124
secondary clustering, 124
universal hashing, 125–127
hash value, 118
header, 70
heap, 99–111
bottom-up construction, 109–111
heap-order property, 99
heap-sort, 107–111, 218
height, 79–80
height-balance property, 152, 154, 157
Helm, 379
Hennessy, 683
hierarchical, 56
Hinrichs, 590
Hirs, 284
Hoare, 256
Hochbaum, 642
Hopcroft, 137, 216, 256, 338, 510, 642, 683
Horner's method, 52
Horner's rule, 488
Hu, 284
Huang, 256
Huffman, 450
Huffman coding, 440–441
Huitema, 544
Ibarra, 642
identity matrix, 492
in-degree, 290
in-place, 248, 324
incidence container, 299
incidence matrix, 612
incident, 290
incoming edges, 290
independent, 28, 30
independent set, 284
INDEPENDENT-SET, 640
index, see discrete logarithm
induction, 25–26
inorder traversal, 146, 150, 155
input size, 593
insertion-sort, 98, 218
inside node, 551
integer multiplication, 270–272
internal memory, 645
inverse shuffle, 502
inversion, 254
IP routers, 514
items, 114
iterative substitution, 264
iterator, 74
Jacobi symbol, 468
Jálá, 137, 683
Jarník, 379
Java Virtual Machine, 64, 137
JDSL, 137
Johnson, 642
Jones, 338
Josephus problem, 133
Karatsuba, 284
Karger, 379
Karp, 137, 414, 642
k-D tree, 563–564
key, 94, 114, 115, 159
key transfer, 475
Kim, 642
Klein, 379, 642
KNAPSACK, 614, 619–621, 634, 639–642
knapsack problem, 259–260, 278–281
Knuth, 54, 137, 216, 256, 338, 450, 683
Komlós, 683
Kosaraju, 338
Koutsoupias, 683
Kruskal, 379, 683
Kruskal’s algorithm, 362–366
L’Hôpital’s Rule, 688
Landis, 216
Langston, 256
language, 594
last node, 99
last-in first-out, 57
LCS, see longest common subsequence
leader election, 517–523
leaves, 75
Lecroq, 450
LEDA, 137
Lee, 590
left child, 76
left subtree, 76
left turn, 574
Legendre symbol, 467
Leighton, 683
Leiserson, 216, 338, 379, 414, 510, 642
level, 84, 313
level numbering, 90
level order traversal, 134
Levin, 642
Lewis, 642
lexicographical, 242
LIFO, 57
Lindholm, 137
line, 572
linear exponential, 688
linear probing, 123
linearity of expectation, 29, 246
link components, 310
link relation, 308
link-state algorithm, 534–535
linked list
doubly linked, 70–73
linked structure, 92
Liskov, 137
list, 68–72, 115
abstract data type, 69–72
list ranking, 661
literals, 605
little-o notation, 18
little-omega notation, 18
live objects, 323
load factor, 122
local replacement, 603
locality-of-reference, 647
locator pattern, 112–113
log file, 115, 121
logarithm, 23, 685
natural, 685
longest common subsequence, 443–446
looking-glass heuristic, 422
lookup table, 142–145
loop invariant, 27
Lovász, 642
LRU, 656
Lynch, 544
Lyons, 683
machine scheduling, 283
Magnanti, 338, 379, 414
main memory, 645
mark-sweep algorithm, 323
Marker strategy, 671
master method, 268–270
matching, 396
matrix chain-product, 274–277
matrix closure, 357–359
matroid theory, 284
Max-Flow, Min-Cut Theorem, 389
maximal independent set, 338
maximum bipartite matching, 396–397
maximum flow, see network flow
McCReight, 450, 590, 683
McDiarmid, 137
McGeoch, 54
median, 245
Megiddo, 256
Mehlhorn, 216, 338, 379, 414, 590, 683
memory hierarchy, 645
memory management, 323, 646–648, 668–673
merge-sort, 219–224
  multi-way, 654–656
tree, 220
mergeable heap, 215
Merzbach, 54
message-passing model, 514–516
method stack, 59–60
minimum cut, 386
minimum spanning tree, 360–372, 528
  Barůvka’s algorithm, 369–372, 528–529
  Kruskal’s algorithm, 362–366, 372
  Prim-Jarnik algorithm, 366–367, 372
minimum-cost flow, 398–405
Minotaur, 288
modular arithmetic, 62, 126, 454, 458–462, 686
modular exponentiation, 462–464
modular multiplicative inverse, 464
modulus, 454
Moore, 450
Morris, 450
Morrison, 450
Motwani, 216, 256, 683
Mount, 590
MST, see minimum spanning tree
multi-way search tree, 159
multi-way tree, 159–162
multicast routing, 535–540
multiplicative group, 461
multiplicative inverse, 458, 464
multiprogramming, 63
Munro, 137
mutually independent, 28

Naimipour, 642
natural logarithm, 685
Neapolitan, 642
network flow, 382–414
  augmenting cycle, 398
  augmenting path, 388
  backward edge, 385
bottleneck, 395
capacity rule, 383
conservation rule, 383
cut, 385–386
cut capacity, 386
dge capacity, 383
Edmonds-Karp algorithm, 393–395
flow across a cut, 386
flow network, 383–385
flow value, 384

Ford-Fulkerson algorithm, 387–395
forward edge, 385
Max-Flow, Min-Cut Theorem, 389
maximum flow problem, 384, 387
minimum cut, 386
minimum-cost flow, 398–405
residual capacity, 387
residual distance, 393
residual graph, 392

network protocol stack, 514
networking protocol stack, 513–514
  application layer, 513
data-link layer, 513
  network layer, 513
physical layer, 513
transport layer, 513
node, 68, 75, 77, 289
ancestor, 75
balanced, 154
boundary, 551
child, 75
descendent, 75
external, 75
inside, 551
internal, 75
outside, 551
parent, 75
redundant, 433
root, 75
sibling, 75
size, 191
unbalanced, 154

nontree edge, 318, 320
NP, 595, 596
NP-completeness, 592–642
NP-hard, 600
null string, 419
number theory, 453–471

O’Rourke, 590
object-oriented design, 137
objective function, 259
Ofman, 284
one-time pad, 474
one-way hash function, 481
online algorithm, 667–679
open addressing, 123, 124
optimization problem, 278, 594
order statistic, 245
orientation, 574, 575
origin, 290
Orlin, 338, 379, 414
orthogonal segments, 565
Ottmann, 590
Index

text continues...
three-sided, 556
range tree, 549–561
range-search query, 549
rank, 142, 231
rank groups, 232
Rao, 510
ratio test, 45
reachability, 316
recurrence equation, 12, 224, 263–270
recursion, 12, 60
recursion tree, 265
recursive doubling, 661
red-black tree, 170–184, 209, 569
  depth property, 170
  external property, 170
  internal property, 170
  recoloring, 174
  root property, 170
Reed, 137
reflexive property, 94, 308
rehashing, 122
Reif, 683
relatively prime, 454, 458
relaxation, 343, 349
renter's dilemma, 667
repeated squaring, 463
rescalable, 619
residual capacity, 387
residual distance, 393
residual graph, 392
residue, 458
restriction, 603
restructure
  trinode, 155
Ribeiro-Neto, 450
right child, 76
right subtree, 76
right turn, 574
Robson, 137
root, 75
root objects, 323
Rosenkrantz, 642
rotation, 155, 157
  double, 155
  single, 155
round-robin, 64
router, 530
routing
  broadcast, 530–531
  center-based trees, 537–538
  distance vector algorithm, 532–533
  link-state algorithm, 534–535
  multicast, 535–540
  reverse path forwarding, 535–536
  unicast, 532–535
RPF, see reverse path forwarding
RSA, 476
RSA cryptosystem, 476–478
running time, 4–6, 10–12
Samet, 590, 683
sample space, 28
SAT, 638
satisfying assignment, 598
Saxe, 284
scan forward, 197
Schaffer, 137
Schönhage, 284
search engine, 225, 439
secondary clustering, 124
security algorithms, 481–487
Sedgewick, 54, 137, 216, 510, 590
seed, 195
segment, 572, 573
segment intersection, 565–568
Seidel, 590
selection, 245–247
selection-sort, 97, 218
self-loop, 291
sentinel, 70, 115
separate chaining, 121
separation edge, 307
separation vertex, 307
sequence, 73–74
  abstract data type, 73–74
  sequential subsets, 659
SET, 484
set, 225–234
SET-COVER, 610, 611, 625, 626, 638, 642
Shamir, 476
Shamos, 590, 683
Shiloach, 683
Shing, 284
shortest path, 341–359
  Bellman-Ford algorithm, 349–351
  Dijkstra's algorithm, 342–348, 373–376
  matrix multiplication, 355–359
sibling, 75
sieve algorithm, 136
sink, 382, 383
skip list, 195–202
  analysis, 200–202
  insertion, 198–199
  levels, 196
  removal, 199–200
  searching, 197–198
towers, 196
update operations, 198–200
Sleator, 216, 683
Solovay-Strassen algorithm, 467
sorting, 96, 218–224, 235–243
bucket-sort, 241–242
external-memory, 654–656
heap-sort, 107–111
in-place, 108, 248
insertion-sort, 98
lower bound, 239–240
merge-sort, 219–224
priority-queue, 96
quick-sort, 235–238
radix-sort, 242–243
selection-sort, 97
stable, 242
source, 382, 383
space usage, 4
spanning forest, 306, 315
spanning subgraph, 292
spanning tree, 293, 303, 305, 306, 313, 315, 316, 360
splay tree, 185–194
split, 164, 653
stable, 242
stack, 57–60
abstract data type, 57
array implementation, 57–58
start time, 261
Steams, 642
Steiglitz, 284, 642
Steiner trees, 538–540
Stephen, 450
Stirling’s Approximation, 686
STL, 137
stop words, 431, 450
Strassen, 284
Strassen’s Algorithm, 272
string
abstract data type, 419–420
null, 419
pattern matching, see pattern matching
prefix, 419
suffix, 419
strongly collision-resistant, 481
strongly connected, 316
strongly NP-hard, 612
subgraph, 292
SUBGRAPH-ISOMORPHISM, 640
subproblem optimality, 275, 278
subproblem overlap, 278
subsequence, 443
SUBSET-SUM, 612–614, 630, 638, 642
substring, 419
subtree, 75
suffix, 419
summation, 21, 687
geometric, 21
symmetric encryption, 473–474
symmetric property, 308
symmetric relation, 289
Szemerédi, 683
Tamassia, 137, 338, 510
Tarjan, 54, 137, 216, 256, 338, 379, 414, 683
task scheduling, 261
Tel, 544
telepathic sum, 35, 687
template method pattern, 332
text compression, 440–441
Theseus, 288
thread, 63–64
timestamping, 482
topological ordering, 325–327
total order, 94
totient function, 460
tower, 198
tower-of-twos, 231
trailer, 70
transfer, 167
transitive closure, 316, 319
transitive property, 94, 308
traveling salesman problem, 342
traveling salesperson problem, 617
treap, 590
tree, 75–137, 292
abstract data type, 77–78
binary, see binary tree
child node, 75
depth, 79–80
external node, 75
height, 79–80
internal node, 75
level, 84
linked structure, 93
multi-way, 159–162
multidimensional, 549–564
node, 75
ordered, 76
parent node, 75
root node, 75
tree edge, 318, 320
tree traversal, 81–83, 86–89
Euler tour, 88–89
generic, 88–89
inorder, 87
level order, 134
postorder, 82–83, 86
preorder, 81, 86
triangle inequality, 623
triangulation, 284
trie, 429–439
  compressed, 433
  standard, 429
trinode restructuring, 155, 173
Tsakalidis, 216
TSP, 617, 619, 623, 624, 633, 641, 642
Tukey, 510
two-dimensional dictionary, 549
(2,4) tree, 163–169
  depth property, 163
  size property, 163
Ullman, 137, 216, 256, 510, 642, 683
unary encoding, 593
underflow, 167, 653
unicast routing, 530, 532–535
union-by-size, 230
union-find, 227–234
universal hashing, 125–127
unordered dictionary, 114
up-heap bubbling, 102
upper envelope, 283
Valiant, 683
van Leeuwen, 338, 379, 414
vector, 65–67, 115, 142
  abstract data type, 65
verification, 596
vertex, 289
  degree, 290
  in-degree, 290
  out-degree, 290
vertex cover, 598
VERTEX-COVER, 599, 608–613, 615, 622, 642
virtual memory, 647
Vishkin, 137, 683
Vismara, 590
Vitter, 54, 683
Voronoi diagram, 589
Vuillemin, 590

Web crawler, 303, 443
Web spider, 303, 443
Wegner, 137
Williams, 137
Wood, 137, 590

X.509, 486