This index uses the following conventions. Numbers are alphabetized as if spelled out; for example, “2-3-4 tree” is indexed as if it were “two-three-four tree.” When an entry refers to a place other than the main text, the page number is followed by a tag: ex. for exercise, pr. for problem, fig. for figure, and n. for footnote. A tagged page number often indicates the first page of an exercise, problem, figure, or footnote; this is not necessarily the page on which the reference actually appears.

\(\alpha(n)\), 511
\(o\)-notation, 47–48
\(O\)-notation, 43 fig., 44–45
\(O^\prime\)-notation, 59 pr.
\(O\)-notation, 59 pr.
\(ω\)-notation, 48
\(Ω\)-notation, 43 fig., 45–46
\(Ω\)-notation, 59 pr.
\(Ω\)-notation, 59 pr.
\(ρ(n)\)-approximation algorithm, 1022
\(Θ\)-notation, 42–44, 43 fig.
\(Θ\)-notation, 59 pr.
\(\{\}\) (set), 1070
\(\in\) (set member), 1070
\(\not\in\) (not a set member), 1070
\(\emptyset\) (empty set), 1070
\(\subseteq\) (subset), 1071
\(\subset\) (proper subset), 1071
\(\colon\) (such that), 1071
\(\cap\) (set intersection), 1071
\(\cup\) (set union), 1071
\(\setminus\) (set difference), 1071
\(|\quad|\) (flow value), 644
\(|\quad|\) (length of a string), 907
\(|\quad|\) (set cardinality), 1073
\times

\((\text{Cartesian product}), 1074\)
\((\text{cross product}), 934\)
\(\{\}\) (sequence), 1078
\((\text{standard encoding}), 975\)
\(\binom{n}{k}\) (choose), 1096
\((\text{factorial}), 54\)
\(\lceil\quad\rceil\) (ceiling), 51
\(\lfloor\quad\rfloor\) (floor), 51
\(\sum\) (sum), 1058
\(\prod\) (product), 1061
\(\rightarrow\) (adjacency relation), 1080
\(\sim\) (reachability relation), 1081
\(\land\) (AND), 633, 987
\(\neg\) (NOT), 987
\(\lor\) (OR), 633, 987
\(\oplus\) (group operator), 862
\(\otimes\) (convolution operator), 825
\(*\) (closure operator), 976
\(\mid\) (divides relation), 850
\(\not\mid\) (does-not-divide relation), 850
\(\equiv\) (equivalent modulo \(n\)), 52, 1077 ex.
\(\not\equiv\) (not equivalent modulo \(n\)), 52
\([a]_{n}\) (equivalence class modulo \(n\)), 851
\(+_{n}\) (addition modulo \(n\)), 863
\(\cdot_{n}\) (multiplication modulo \(n\)), 863
\(\left(\right)\) (Legendre symbol), 903 pr.
ε (empty string), 907, 976
□ (prefix relation), 907
■ (suffix relation), 907
\( \geq_x \) (above relation), 941
▷ (comment symbol), 19
\( \leq_P \) (polynomial-time reducibility relation), 984, 994 ex.

AA-tree, 301
abelian group, 862
ABOVE, 942
above relation, 941
absent child, 1089
absolutely convergent series, 1059
absorption laws for sets, 1072
abstract problem, 972
acceptable pair of integers, 894
acceptance
by an algorithm, 976
by a finite automaton, 916
accepting state, 916
accounting method, 410–412
for binary counters, 411–412
for dynamic tables, 419
for stack operations, 410–411, 412 ex.
Ackermann’s function, 521
activity-selection problem, 371–379
acyclic graph, 1082
relation to matroids, 403 pr.
add instruction, 22
addition
of binary integers, 21 ex.
of matrices, 728
modulo \( n \) \((\mod n)\), 863
of polynomials, 822
additive group modulo \( n \), 863
addressing, open, see open addressing
adjacency-list representation, 528
adjacency-matrix representation, 529
adjacent vertices, 1080
admissible edge, 681
admissible network, 681–683
aggregate analysis, 406–410
for binary counters, 408–409
for breadth-first search, 534
for depth-first search, 542–543
for Dijkstra’s algorithm, 598
for disjoint-set data structures, 503–504, 505 ex., 509 ex.

for dynamic tables, 418–419
for Fibonacci heaps, 493 ex.
for Graham’s scan, 954–955
for shortest paths in a dag, 592
for stack operations, 406–408
aggregate flow, 788
Akra-Bazzi method for solving a recurrence, 89–90
AKS sorting network, 724
algorithm, 5
correctness of, 6
origin of word, 40
running time of, 23
as a technology, 12
Alice, 881
ALLOCATE-NODE, 442
ALLOCATE-OBJECT, 211
allocation of objects, 210–212
all-pairs shortest paths, 581, 620–642
in \( \epsilon \)-dense graphs, 641 pr.
Floyd-Warshall algorithm for, 629–632
Johnson’s algorithm for, 636–640
by matrix multiplication, 622–629
by repeated squaring, 625–627
alphabet, 916, 975
\( \alpha(n) \), 511
amortized analysis, 405–429
accounting method of, 410–412
aggregate analysis, 406–410
for bit-reversal permutation, 425 pr.
for breadth-first search, 534
for depth-first search, 542–543
for Dijkstra’s algorithm, 598
for disjoint-set data structures, 503–504, 505 ex., 509 ex., 512–517, 518 ex.
for dynamic tables, 416–425
for the generic push-relabel algorithm, 678–679
for Graham’s scan, 954–955
for the Knuth-Morris-Pratt algorithm, 926–927
for making binary search dynamic, 426 pr.
potential method of, 412–416
for restructuring red-black trees, 428 pr.
for shortest paths in a dag, 592
for stacks on secondary storage, 452 pr.
for weight-balanced trees, 427 pr.
amortized cost
in the accounting method, 410
in aggregate analysis, 406
in the potential method, 413
analysis of algorithms, 21–27
see also amortized analysis, probabilistic analysis
ancestor, 1087
least common, 521 pr.
AND function (\∧), 633, 987
AND gate, 987
and, in pseudocode, 20
antisymmetry, 1076
ANY-SEGMENTS-INTERSECT, 943
approximation
by least squares, 762–765
of summation by integrals, 1067
approximation algorithm, 1022–1054
for bin packing, 1049 pr.
for MAX-CNF satisfiability problem, 1043 ex.
for MAX-CUT problem, 1043 ex.
for the maximum-clique problem, 1050 pr.
for maximum matching, 1051 pr.
for MAX-3-CNF satisfiability, 1039–1040
for minimum-weight vertex cover, 1040–1043
for parallel-machine-scheduling problem, 1051 pr.
randomized, 1039
for the set-covering problem, 1033–1038
for the subset-sum problem, 1043–1049
for the traveling-salesman problem, 1027–1033
for the vertex-cover problem, 1024–1027
for the weighted set-covering problem, 1050 pr.
approximation ratio, 1022, 1039
approximation scheme, 1023
APPROX-MIN-WEIGHT-VC, 1041
APPROX-SUBSET-SUM, 1046
APPROX-TSP-TOUR, 1028
APPROX-VERTEX-COVER, 1025
arbitrage, 615 pr.
arb, see edge
argument of a function, 1078
arithmetic, modular, 51–52, 862–869
arithmetic instructions, 22
arithmetic series, 1059
arithmetic with infinities, 587
arm, 435
array, 19
Monge, 88 pr.
articulation point, 558 pr.
assembly-line scheduling, 324–331
assignment
multiple, 19
satisfying, 988, 996
truth, 988, 996
associative laws for sets, 1072
associative operation, 862
asymptotically larger, 49
asymptotically nonnegative, 42
asymptotically positive, 42
asymptotically smaller, 49
asymptotically tight bound, 42
asymptotic efficiency, 41
asymptotic lower bound, 45
asymptotic notation, 41–50, 59 pr.
and graph algorithms, 526
and linearity of summations, 1059
asymptotic upper bound, 44
attribute of an object, 20
augmenting data structures, 302–318
augmenting path, 654, 696 pr.
authentication, 251 pr.
automaton
finite, 916
string-matching, 917–922
auxiliary hash function, 239
auxiliary linear program, 811
average-case running time, 26
AVL-INSERT, 296 pr.
AVL tree, 296 pr.
axioms, for probability, 1100
back edge, 546, 550
back substitution, 745
bad guy, 539 ex.
BAD-SET-COVER-INSTANCE, 1038 ex.
BALANCE, 296 pr.
balanced search tree
AA-trees, 301
AVL trees, 296 pr.
B-trees, 434–454
k-neighbor trees, 301
red-black trees, 273–301
scapegoat trees, 301
splay trees, 301, 432
treaps, 296 pr.
2-3-4 trees, 439, 453 pr.
2-3 trees, 300, 454
weight-balanced trees, 301, 427 pr.
bases, 350
base-a pseudoprime, 889
base case, 64
basic feasible solution, 792
basic solution, 792
basic variable, 782
basis function, 762
Batcher’s odd-even merging network, 721 pr.
Bayes’s theorem, 1104
Bellman-Ford, 588
Bellman-Ford algorithm, 588–592
for all-pairs shortest paths, 620, 639
in Johnson’s algorithm, 639
and objective functions, 606–607 ex.
to solve systems of difference constraints, 605
Yen’s improvement to, 614 pr.
Below, 942
Bernoulli trial, 1112
and balls and bins, 109–110
and streaks, 110–114
best-case running time, 27 ex., 46
BFS, 532
Biased-Random, 94 ex.
biconnected component, 558 pr.
bigo notation, 43 fig., 44–45
big-omega notation, 43 fig., 45–46
bijective function, 1079
binary character code, 385
binary counter
analyzed by accounting method, 411–412
analyzed by aggregate analysis, 408–409
analyzed by potential method, 414–415
and binomial heaps, 472 ex.
bit-reversed, 425 pr.
binary entropy function, 1098
binary gcd algorithm, 902 pr.
binary heap, see heap
binary relation, 1075
binary search, 37 ex.
with fast insertion, 426 pr.
in insertion sort, 37 ex.
in searching B-trees, 449 ex.
binary search trees, 253–272
AA-trees, 301
AVL trees, 296 pr.
deletion from, 262–263
with equal keys, 268 pr.
insertion into, 261
k-neighbor trees, 301
maximum key of, 258
minimum key of, 258
optimal, 356–363, 369
predecessor in, 258–259
querying, 256–261
randomly built, 265–268, 270 pr.
scapegoat trees, 301
searching, 256–258
for sorting, 264 ex.
splay trees, 301
successor in, 258–259
and treaps, 296 pr.
weight-balanced trees, 301
see also red-black tree
binary-search-tree property, 254
vs. min-heap property, 256 ex.
binary tree, 1088
full, 1089
number of different ones, 271 pr.
representation of, 214
see also binary search tree
binomial coefficient, 1096–1098
binomial distribution, 1113–1116
and balls and bins, 109
maximum value of, 1117 ex.
tails of, 1118–1125
binomial expansion, 1096
binomial heap, 455–475
and binary counter and binary addition, 472 ex.
creating, 461
decreasing a key in, 470
deletion from, 470–471
extracting the minimum key from, 468–469
insertion into, 468
minimum key of, 462
in minimum-spanning-tree algorithm, 474 pr.
properties of, 459
running times of operations on, 456 fig.
uniting, 462–468
Binomial-Heap-Decrease-Key, 470
Binomial-Heap-Delete, 470
<table>
<thead>
<tr>
<th><strong>Index</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Binomial-Heap-Extract-Min, 468</td>
</tr>
<tr>
<td>Binomial-Heap-Insert, 468</td>
</tr>
<tr>
<td>Binomial-Heap-Merge, 464</td>
</tr>
<tr>
<td>Binomial-Heap-Minimum, 462</td>
</tr>
<tr>
<td>Binomial-Heap-Union, 463</td>
</tr>
<tr>
<td>Binomial-Link, 462</td>
</tr>
<tr>
<td>Binomial tree, 457–459</td>
</tr>
<tr>
<td>unordered, 479</td>
</tr>
<tr>
<td>Bin packing, 1049 pr.</td>
</tr>
<tr>
<td>Bipartite graph, 1083</td>
</tr>
<tr>
<td>corresponding flow network of, 665</td>
</tr>
<tr>
<td>and hypergraphs, 1084 ex.</td>
</tr>
<tr>
<td>Bipartite matching, 497, 664–669</td>
</tr>
<tr>
<td>Hopcroft-Karp algorithm for, 696 pr.</td>
</tr>
<tr>
<td>Birthday paradox, 106–109</td>
</tr>
<tr>
<td>Biscuit to be kept out of basket, 646</td>
</tr>
<tr>
<td>Bisection of a tree, 1092 pr.</td>
</tr>
<tr>
<td>Bitonic euclidean traveling-salesman problem, 364 pr.</td>
</tr>
<tr>
<td>Bitonic sequence, 712</td>
</tr>
<tr>
<td>and shortest paths, 618 pr.</td>
</tr>
<tr>
<td>Bitonic-Sorter, 715</td>
</tr>
<tr>
<td>Bitonic sorting network, 712–716</td>
</tr>
<tr>
<td>Bitonic tour, 364 pr.</td>
</tr>
<tr>
<td>Bit operation, 850</td>
</tr>
<tr>
<td>in Euclid’s algorithm, 902 pr.</td>
</tr>
<tr>
<td>Bit-reversal permutation, 425 pr., 841</td>
</tr>
<tr>
<td>Bit-Reverse-Copy, 842</td>
</tr>
<tr>
<td>Bit-reversed binary counter, 425 pr.</td>
</tr>
<tr>
<td>Bit-Reversed-Increment, 425 pr.</td>
</tr>
<tr>
<td>Bit vector, 222 ex.</td>
</tr>
<tr>
<td>Black-height, 274</td>
</tr>
<tr>
<td>Black vertex, 531, 540</td>
</tr>
<tr>
<td>Blocking flow, 697</td>
</tr>
<tr>
<td>Block structure in pseudocode, 19</td>
</tr>
<tr>
<td>Bob, 881</td>
</tr>
<tr>
<td>Boole's inequality, 1105 ex.</td>
</tr>
<tr>
<td>Boolean combinational circuit, 988</td>
</tr>
<tr>
<td>Boolean combinational element, 987</td>
</tr>
<tr>
<td>Boolean connective, 996</td>
</tr>
<tr>
<td>Boolean formula, 967, 983 ex., 996, 1002 ex.</td>
</tr>
<tr>
<td>Boolean function, 1098 ex.</td>
</tr>
<tr>
<td>Boolean matrix multiplication</td>
</tr>
<tr>
<td>and transitive closure, 759 ex.</td>
</tr>
<tr>
<td>Borůvka’s algorithm, 578</td>
</tr>
<tr>
<td>Bottleneck spanning tree, 577 pr.</td>
</tr>
<tr>
<td>Bottleneck traveling-salesman problem, 1033 ex.</td>
</tr>
<tr>
<td>Bottom of a stack, 200</td>
</tr>
<tr>
<td>Bound</td>
</tr>
<tr>
<td>asymptotically tight, 42</td>
</tr>
<tr>
<td>asymptotic lower, 45</td>
</tr>
<tr>
<td>asymptotic upper, 44</td>
</tr>
<tr>
<td>on binomial coefficients, 1097–1098</td>
</tr>
<tr>
<td>on binomial distributions, 1116</td>
</tr>
<tr>
<td>polylogarithmic, 54</td>
</tr>
<tr>
<td>on the tails of a binomial distribution, 1118–1125</td>
</tr>
<tr>
<td>Boundary condition, 63–64</td>
</tr>
<tr>
<td>Boundary of a polygon, 939 ex.</td>
</tr>
<tr>
<td>Bounding a summation, 1062–1069</td>
</tr>
<tr>
<td>Box, nesting, 615 pr.</td>
</tr>
<tr>
<td>B*-tree, 438</td>
</tr>
<tr>
<td>Branching factor, in B-trees, 437</td>
</tr>
<tr>
<td>Branch instructions, 22</td>
</tr>
<tr>
<td>Breadth-first search, 531–539</td>
</tr>
<tr>
<td>and shortest paths, 534–537, 581</td>
</tr>
<tr>
<td>Similarity to Dijkstra’s algorithm, 599, 600 ex.</td>
</tr>
<tr>
<td>Breadth-first tree, 532, 538</td>
</tr>
<tr>
<td>Bridge, 558 pr.</td>
</tr>
<tr>
<td>B*-tree, 439</td>
</tr>
<tr>
<td>B-tree, 434–454</td>
</tr>
<tr>
<td>Creating, 442</td>
</tr>
<tr>
<td>Deletion from, 449–452</td>
</tr>
<tr>
<td>Full node in, 439</td>
</tr>
<tr>
<td>Height of, 439–440</td>
</tr>
<tr>
<td>Insertion into, 443–447</td>
</tr>
<tr>
<td>Minimum degree of, 439</td>
</tr>
<tr>
<td>Minimum key of, 447 ex.</td>
</tr>
<tr>
<td>Properties of, 438–441</td>
</tr>
<tr>
<td>Searching, 441–442</td>
</tr>
<tr>
<td>Splitting a node in, 443–445</td>
</tr>
<tr>
<td>2-3-4 trees, 439</td>
</tr>
<tr>
<td>B-Tree-Create, 442</td>
</tr>
<tr>
<td>B-Tree-Delete, 449</td>
</tr>
<tr>
<td>B-Tree-Insert, 445</td>
</tr>
<tr>
<td>B-Tree-Insert-Nonfull, 446</td>
</tr>
<tr>
<td>B-Tree-Search, 442, 449 ex.</td>
</tr>
<tr>
<td>B-Tree-Split-Child, 444</td>
</tr>
<tr>
<td>Bubblesort, 38 pr.</td>
</tr>
<tr>
<td>Bucket, 174</td>
</tr>
<tr>
<td>Bucket sort, 174–177</td>
</tr>
<tr>
<td>Bucket-Sort, 174</td>
</tr>
<tr>
<td>Build-Max-Heap, 133</td>
</tr>
<tr>
<td>Build-Max-Heap', 142 pr.</td>
</tr>
<tr>
<td>Build-Min-Heap, 135</td>
</tr>
</tbody>
</table>
butterfly operation, 839

cache, 22

cache-oblivious, 454
call

a subroutine, 22, 23 n.
by value, 20
cancellation lemma, 831
cancellation of flow, 646
canonical form for task scheduling, 399
capacity

of a cut, 655
of an edge, 644
residual, 651, 654
capacity constraint, 644
cardinality of a set (| |), 1073
Carmichael number, 890, 896 ex.
Cartesian product (×), 1074
Cartesian sum, 830 ex.
cascading cut, 490
CASCADING-CUT, 490
Catalan numbers, 271 pr., 333
ceiling function (⌈⌉), 51
in master theorem, 81–84
ceiling instruction, 22
certain event, 1100
certificate

in a cryptosystem, 886
for verification algorithms, 980
CHAINED-HASH-DELETE, 226
CHAINED-HASH-INSERT, 226
CHAINED-HASH-SEARCH, 226
chaining, 225–228, 250 pr.
chain of a convex hull, 955
changing

a key in a Fibonacci heap, 497 pr.
variables in the substitution method, 66
character code, 385
child, 1087, 1089
child list in a Fibonacci heap, 477
Chinese remainder theorem, 873–876
chirp transform, 838 ex.
choose ( ), 1096
chord, 308 ex.
ciphertext, 883
circuit

boolean combinational, 988
for Fast Fourier Transform, 842–843
CIRCUIT-SAT, 989
circuit satisfiability, 987–994
circular, doubly linked list with a sentinel, 206
circular linked list, 204
see also linked list
class

complexity, 977
equivalence, 1075
classification of edges

in breadth-first search, 558 pr.
in depth-first search, 546–547, 548 ex.
clause, 999
clean sequence, 713
clique, 1003
CLIQUE, 1003
clique problem, 1003–1006
approximation algorithm for, 1050 pr.
closed interval, 311
closed semiring, 642
closest pair, finding, 957–962
closest-point heuristic, 1033 ex.
closure

group property, 862
of a language, 976
operator (·), 976
transitive, see transitive closure
clustering, 239–240
CNF (conjunctive normal form), 967, 999
CNF satisfiability, 1043 ex.
code, 385
Huffman, 385–393
codomain, 1077
coefficient

binomial, 1096
of a polynomial, 52, 822
in slack form, 783
coefficient representation, 824
and fast multiplication, 827–829
cofactor, 732
coin changing, 402 pr.
collinearity, 935
collision, 224
resolution by chaining, 225–228
resolution by open addressing, 237–245
coloring, 1019 pr., 1091 pr.
color of a red-black-tree node, 273
column rank, 731
column vector, 726
combinational circuit, 988
combinational element, 987
Index

comment in pseudocode (\textellipsis{}), 19
commodity, 788
common divisor, 852
greatest, see greatest common divisor
common multiple, 861 ex.
common subexpression, 839
common subsequence, 351
longest, 350–356, 369
commutative laws for sets, 1071
commutativity under an operator, 862
	extsc{Compactify-List}, 213 ex.
compact list, 218 pr.
	extsc{Compact-List-Search}, 218 pr.
	extsc{Compact-List-Search’}, 219 pr.
comparable line segments, 941
comparator, 705
comparison network, 704–709
comparison sort, 165
and binary search trees, 256 ex.
and mergeable heaps, 489 ex.
randomized, 178 pr.
and selection, 192
compatible activities, 371
compatible matrices, 332, 729
complement
of an event, 1101
of a graph, 1007
of a language, 976
Schur, 748, 761
of a set, 1072
complementary slackness, 818 pr.
complete graph, 1083
complete \( k \)-ary tree, 1090
see also heap
completeness of a language, 994 ex.
completion time, 402 pr., 1051 pr.
complexity class, 977
co-NP, 982
NP, 967, 981
NPC, 968, 986
P, 967, 973
complexity measure, 977
complex numbers, multiplication of, 741 ex.
complex root of unity, 830
interpolation at, 836–837
component
biconnected, 558 pr.
connected, 1082
strongly connected, 1082
component graph, 554
composite number, 851
witness to, 890
computational geometry, 933–965
computational problem, 5–6
	extsc{Compute-Prefix-Function}, 926
	extsc{Compute-Transition-Function}, 922
concatenation
of languages, 976
of strings, 907
concrete problem, 973
conditional branch instruction, 22
conditional independence, 1106 ex.
conditional probability, 1103–1104
configuration, 991
conjugate transpose, 759 ex.
conjunctive normal form, 967, 999
connected component, 1082
identified using depth-first search, 549 ex.
identified using disjoint-set data structures, 499–501
	extsc{Connected-Components}, 500
connected graph, 1082
connective, 996
cos-NP, 982
conservation of flow, 644
consistency of literals, 1004
	extsc{Consolidate}, 486
consolidating a Fibonacci-heap root list, 483
constraint, 777
difference, 602
equality, 606 ex., 778, 780
inequality, 778, 780
linear, 772
nonnegativity, 777, 779
tight, 791
violation of, 791
constraint graph, 603–605
contain, in a path, 1081
continuous uniform probability distribution, 1102
contraction
of a dynamic table, 420–422
of a matroid, 397
of an undirected graph by an edge, 1084
control instructions, 22
convergence property, 587, 609
convergent series, 1059
convex combination of points, 934
### Index

- **convex function**, 1109
- **convex hull**, 947–957, 964 pr.
- **convex layers**, 962 pr.
- **convex polygon**, 939 ex.
- **convex set**, 650 ex.
- **convolution** \((\otimes)\), 825
- **convolution theorem**, 837
- **copy instruction**, 22
- **correctness of an algorithm**, 6
- **countably infinite set**, 1073
- **counter**, see **binary counter**
- **counting**, 1094–1100
  - **probabilistic**, 118 pr.
- **counting sort**, 168–170
  - **in radix sort**, 172
- **COUNTING-SORT**, 168
- **coupon collector’s problem**, 110
- **cover**
  - **path**, 692 pr.
  - **by a subset**, 1034
  - **vertex**, 1006, 1024, 1040–1043
- **covertical**, 942
- **credit**, 410
- **critical edge**, 662
- **critical path**, 594
- **cross edge**, 546
- **crossing a cut**, 563
- **cross product** \((\times)\), 934
- **cryptosystem**, 881–887
- **cubic spline**, 767 pr.
- **curve fitting**, 762–765
- **cut**
  - **cascading**, 490
  - **of a flow network**, 654–657
  - **of an undirected graph**, 563
  - **weight of**, 1043 ex.
- **CUT**, 489
- **cutting**, in a **Fibonacci heap**, 490
- **cycle of a graph**, 1081–1082
  - **hamiltonian**, 967, 979
  - **minimum mean-weight**, 617 pr.
  - **negative-weight**, see **negative-weight cycle**
  - **and shortest paths**, 583
- **cyclic group**, 877
- **cyclic rotation**, 930 ex.
- **cycling**, of simplex algorithm, 802
- **cylinder**, 435

---

<table>
<thead>
<tr>
<th>Dag-Sort-Paths</th>
<th>592</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dag-Shortest-Paths</strong>, 592</td>
<td></td>
</tr>
<tr>
<td><strong>d-ary heap</strong>, 143 pr.</td>
<td></td>
</tr>
<tr>
<td><strong>in shortest-paths algorithms</strong>, 641 pr.</td>
<td></td>
</tr>
<tr>
<td><strong>data-movement instructions</strong>, 22</td>
<td></td>
</tr>
<tr>
<td><strong>data structure</strong>, 8, 197–318, 431–522</td>
<td></td>
</tr>
<tr>
<td><strong>AA-trees</strong>, 301</td>
<td></td>
</tr>
<tr>
<td><strong>augmentation of</strong>, 302–318</td>
<td></td>
</tr>
<tr>
<td><strong>AVL trees</strong>, 296 pr.</td>
<td></td>
</tr>
<tr>
<td><strong>binary search trees</strong>, 253–272</td>
<td></td>
</tr>
<tr>
<td><strong>binomial heaps</strong>, 455–475</td>
<td></td>
</tr>
<tr>
<td><strong>bit vectors</strong>, 222 ex.</td>
<td></td>
</tr>
<tr>
<td><strong>B-trees</strong>, 434–454</td>
<td></td>
</tr>
<tr>
<td><strong>deques</strong>, 204 ex.</td>
<td></td>
</tr>
<tr>
<td><strong>dictionaries</strong>, 197</td>
<td></td>
</tr>
<tr>
<td><strong>direct-address tables</strong>, 222–223</td>
<td></td>
</tr>
<tr>
<td><strong>for disjoint sets</strong>, 498–522</td>
<td></td>
</tr>
<tr>
<td><strong>for dynamic graphs</strong>, 433</td>
<td></td>
</tr>
<tr>
<td><strong>dynamic sets</strong>, 197–199</td>
<td></td>
</tr>
<tr>
<td><strong>dynamic trees</strong>, 432</td>
<td></td>
</tr>
<tr>
<td><strong>exponential search trees</strong>, 182, 433</td>
<td></td>
</tr>
<tr>
<td><strong>Fibonacci heaps</strong>, 476–497</td>
<td></td>
</tr>
<tr>
<td><strong>fusion trees</strong>, 182, 433</td>
<td></td>
</tr>
<tr>
<td><strong>hash tables</strong>, 224–229</td>
<td></td>
</tr>
<tr>
<td><strong>heaps</strong>, 127–144</td>
<td></td>
</tr>
<tr>
<td><strong>interval trees</strong>, 311–317</td>
<td></td>
</tr>
<tr>
<td><strong>k-neighbor trees</strong>, 301</td>
<td></td>
</tr>
<tr>
<td><strong>linked lists</strong>, 204–209</td>
<td></td>
</tr>
<tr>
<td><strong>order-statistic trees</strong>, 302–308</td>
<td></td>
</tr>
<tr>
<td><strong>persistent</strong>, 294 pr., 432</td>
<td></td>
</tr>
<tr>
<td><strong>potential of</strong>, 413</td>
<td></td>
</tr>
<tr>
<td><strong>priority queues</strong>, 138–142</td>
<td></td>
</tr>
<tr>
<td><strong>queues</strong>, 200–203</td>
<td></td>
</tr>
<tr>
<td><strong>radix trees</strong>, 269 pr.</td>
<td></td>
</tr>
<tr>
<td><strong>red-black trees</strong>, 273–301</td>
<td></td>
</tr>
<tr>
<td><strong>relaxed heaps</strong>, 497</td>
<td></td>
</tr>
<tr>
<td><strong>rooted trees</strong>, 214–217</td>
<td></td>
</tr>
<tr>
<td><strong>scapegoat trees</strong>, 301</td>
<td></td>
</tr>
<tr>
<td><strong>on secondary storage</strong>, 434–437</td>
<td></td>
</tr>
<tr>
<td><strong>skip lists</strong>, 301</td>
<td></td>
</tr>
<tr>
<td><strong>splay trees</strong>, 301, 432</td>
<td></td>
</tr>
<tr>
<td><strong>stacks</strong>, 200–201</td>
<td></td>
</tr>
<tr>
<td><strong>treaps</strong>, 296 pr.</td>
<td></td>
</tr>
<tr>
<td><strong>2-3-4 heaps</strong>, 473 pr.</td>
<td></td>
</tr>
<tr>
<td><strong>2-3-4 trees</strong>, 439, 453 pr.</td>
<td></td>
</tr>
<tr>
<td><strong>2-3 trees</strong>, 300, 454</td>
<td></td>
</tr>
<tr>
<td><strong>van Emde Boas</strong>, 433</td>
<td></td>
</tr>
<tr>
<td><strong>weight-balanced trees</strong>, 301</td>
<td></td>
</tr>
<tr>
<td><strong>deadline</strong>, 399</td>
<td></td>
</tr>
<tr>
<td><strong>deallocation of objects</strong>, 210–212</td>
<td></td>
</tr>
</tbody>
</table>
Index

1153

decision by an algorithm, 976
decision problem, 969, 972
  and optimization problems, 969
decision tree, 166–167
  zero-one principle for, 712 ex.
DECREASE-KEY, 138, 455
decreasing a key
  in binomial heaps, 470
  in Fibonacci heaps, 489–492
  in 2-3-4 heaps, 473 pr.
DECREMENT, 409 ex.
degeneracy, 802
degree
  of a binomial-tree root, 457
  maximum, of a Fibonacci heap, 479, 488 ex., 493–496
  minimum, of a B-tree, 439
  of a node, 1088
  of a polynomial, 52, 822
  of a vertex, 1081
degree-bound, 822
DELETE, 198, 455
DELETE-LARGER-HALF, 416 ex.
deletion
  from binary search trees, 262–263
  from binomial heaps, 470–471
  from B-trees, 449–452
  from chained hash tables, 226
  from direct-address tables, 222
  from dynamic tables, 422
  from Fibonacci heaps, 492, 496 pr.
  from heaps, 142 ex.
  from interval trees, 313
  from linked lists, 206
  from open-address hash tables, 238
  from order-statistic trees, 307
  from queues, 201
  from red-black trees, 288–294
  from stacks, 200
  from sweep-line statuses, 942
  from 2-3-4 heaps, 473 pr.
demand paging, 22
DeMorgan’s laws, 1072
dense graph, 527
  $\epsilon$-dense, 641 pr.
density of prime numbers, 887–888
dependence
  and indicator random variables, 96
  linear, 731

see also independence
depth
  average, of a node in a randomly built binary search tree, 270 pr.
  of a comparison network, 707
  of a node in a rooted tree, 1088
  of quicksort recursion tree, 153 ex.
  of SORTER, 720 ex.
  of a sorting network, 708 ex.
  of a stack, 162 pr.
depth-determination problem, 519 pr.
depth-first forest, 540
depth-first search, 540–549
  in finding articulation points, bridges, and biconnected components, 558 pr.
  in finding strongly connected components, 552–557
  in topological sorting, 549–552
depth-first tree, 540
deque, 204 ex.
DEQUEUE, 203
derivative of series, 1060
descent, 1087
destination vertex, 581
det, see determinant
determinant, 732
  and matrix multiplication, 759 ex.
deterministic, 99
DETERMINISTIC-SEARCH, 118 pr.
DFS, 541
DFS-VISIT, 541
DFT (Discrete Fourier Transform), 833
diagonal matrix, 726
  LUP decomposition of, 754 ex.
diameter of a tree, 539 ex.
dictionary, 197
difference constraints, 601–607
difference equation, see recurrence
difference of sets (−), 1071
  symmetric, 696 pr.
differentiation of series, 1060
digital signature, 883
digraph, see directed graph
DIJKSTRA, 595
Dijkstra’s algorithm, 595–601
  for all-pairs shortest paths, 620, 639
  implemented with a Fibonacci heap, 599
  implemented with a min-heap, 599
  with integer edge weights, 600–601 ex.
in Johnson’s algorithm, 639
similarity to breadth-first search, 599, 600 ex.
similarity to Prim’s algorithm, 570, 599
DIRECT-ADDRESS-DELETE, 222
direct addressing, 222–223
DIRECT-ADDRESS-INSERT, 222
DIRECT-ADDRESS-SEARCH, 222
direct-address table, 222–223
directed acyclic graph (dag), 1083
and back edges, 550
and component graphs, 554
and hamiltonian-path problem, 983 ex.
single-source shortest-paths algorithm for, 592–595
topological sort of, 549–552
directed graph, 1080
all-pairs shortest paths in, 620–642
constraint graph, 603
Euler tour of, 559 pr., 966
hamiltonian cycle of, 967
and longest paths, 966
path cover of, 692 pr.
PERT chart, 594, 594 ex.
semiconnected, 557 ex.
shortest path in, 580
single-source shortest paths in, 580–619
singly connected, 549 ex.
square of, 530 ex.
transitive closure of, 632
transpose of, 530 ex.
see also circuit, directed acyclic graph, graph, network
directed segment, 934–935
directed version of an undirected graph, 1082
DIRECTION, 937
Discharge, 683
discharge of an overflowing vertex, 683
discovered vertex, 531, 540
discovery time, in depth-first search, 541
Discrete Fourier Transform, 833
discrete logarithm, 877
discrete logarithm theorem, 877
discrete probability distribution, 1101
discrete random variable, 1106–1111
disjoint-set data structure, 498–522
analysis of, 512–517, 518 ex.
in depth determination, 519 pr.
disjoint-set-forest implementation of, 505–509
in Kruskal’s algorithm, 569
linear-time special case of, 522
linked-list implementation of, 501–505
in off-line least common ancestors, 521 pr.
in off-line minimum, 518 pr.
in task scheduling, 404 pr.
disjoint-set forest, 505–509
analysis of, 512–517, 518 ex.
rank properties of, 511–512, 518 ex.
see also disjoint-set data structure
disjoint sets, 1073
disjunctive normal form, 1000
disk, 947 ex.
see also secondary storage
DISK-READ, 437
DISK-WRITE, 437
distance
edit, 364 pr.
euclidean, 957
$L_m$, 962 ex.
Manhattan, 194 pr., 962 ex.
of a shortest path, 534
distribution
binomial, 1113–1116
geometric, 1112
of inputs, 93, 99
of prime numbers, 888
probability, 1100–1102
sparse-hulled, 964 pr.
distributive laws for sets, 1072
divergent series, 1059
divide-and-conquer method, 28–33
analysis of, 32–33
for binary search, 37 ex.
for conversion of binary to decimal, 856 ex.
for Fast Fourier Transform, 834–836
for finding the closest pair of points, 958–961
for finding the convex hull, 948
for matrix inversion, 756–758
for merge sort, 28–36
for multiplication, 844 pr.
for quicksort, 145–164
and recursion trees, 67
relation to dynamic programming, 323
for selection, 185–193
solving recurrences for, 62–90
Index

for Strassen’s algorithm, 735–742
divide instruction, 22
divides relation (l), 850
division method, 230–231
division theorem, 851
divisor, 850–851
common, 852
see also greatest common divisor
DNA, 350, 364 pr.
DNF (disjunctive normal form), 1000
does-not-divide relation (l), 850
domain, 1077
dominiates relation, 962 pr.
double hashing, 240–241, 244 ex.
doubly linked list, 204
see also linked list
d-regular graph, 669 ex.
duality, 804–811
weak, 805
dual linear program, 805
dynamic graph, 499 n.
data structures for, 433
minimum-spanning-tree algorithm for,
574 ex.
transitive closure of, 641 pr.
dynamic order statistics, 302–308
dynamic-programming method, 323–369
for activity selection, 378 ex.
for all-pairs shortest paths, 622–632
for assembly-line scheduling, 324–331
for bitonic euclidean traveling-salesman
problem, 364 pr.
compared to greedy algorithms, 341,
350 ex., 373–375, 380, 382–383
for edit distance, 364 pr.
elements of, 339–350
for Floyd-Warshall algorithm, 629–632
for longest common subsequence, 350–356
for matrix-chain multiplication, 331–339
and memoization, 347–349
for optimal binary search trees, 356–363
optimal substructure in, 339–344
overlapping subproblems in, 344–346
for printing neatly, 364 pr.
reconstructing an optimal solution in,
346–347
for scheduling, 369 pr.
for transitive closure, 632–635
for Viterbi algorithm, 367 pr.
for 0-1 knapsack problem, 384 ex.
dynamic set, 197–199
see also data structure
dynamic table, 416–425
analyzed by accounting method, 419
analyzed by aggregate analysis, 418–419
analyzed by potential method, 419–420,
422–424
load factor of, 417
dynamic tree, 432
e, 53
E [ ] (expected value), 1108
eyear-first form, 399
eyear task, 399
eedge, 1080
admissible, 681
back, 546
bridge, 558 pr.
capacity of, 644
classification in breadth-first search, 558 pr.
classification in depth-first search, 546–547
critical, 662
cross, 546
forward, 546
inadmissible, 681
light, 563
negative-weight, 582–583
residual, 652
safe, 562
saturated, 672
tree, 538, 540, 546
weight of, 529
degree connectivity, 664 ex.
degree set, 1080
degree distance, 364 pr.
Edmonds-Karp algorithm, 660–663
elementary event, 1100
element of a set (e), 1070
ellipsoid algorithm, 776
elliptic-curve factorization method, 905
e else, in pseudocode, 19
empty language (\), 976
empty set (\), 1070
empty set laws, 1071
empty stack, 201
empty string (e), 907, 975
empty tree, 1089
encoding of problem instances, 973–975
endpoint
  of an interval, 311
  of a line segment, 934
ENQUEUE, 203
entering a vertex, 1080
entering variable, 793
entropy function, 1098
ε-dense graph, 641 pr.
equality
  of functions, 1078
  linear, 772
  of sets, 1070
equality constraint, 606 ex., 778
  and inequality constraints, 780
tight, 791
violation of, 791
equation
  and asymptotic notation, 46–47
  normal, 764
  recurrence, see recurrence
equivalence, modular (≡), 52, 1077 ex.
equivalence class, 1075
  modulo n ([a]n), 851
equivalence relation, 1075–1076
  and modular equivalence, 1077 ex.
equivalent linear programs, 779
escape problem, 692 pr.
esential term, 738
EUCLID, 858
Euclid’s algorithm, 856–862, 902 pr.
euclidean distance, 957
euclidean norm, 730
Euler’s phi function, 865
Euler’s theorem, 877, 896 ex.
Euler tour, 559 pr., 966
  and hamiltonian cycles, 966
ε-universal hash function, 236 ex.
evaluation of a polynomial, 39 pr., 824, 829 ex.
  and its derivatives, 845 pr.
  at multiple points, 846 pr.
event, 1100
  point, 942
  point schedule, 942
EXACT-SUBSET-SUM, 1045
excess flow, 669
exchange property, 393
exclusion and inclusion, 1074 ex.
execute a subroutine, 23 n.
extraction of a dynamic table, 417–418
expectation, see expected value
expected running time, 26
expected value, 1108–1109
  of a binomial distribution, 1114
  of a geometric distribution, 1112
  of an indicator random variable, 95
explored vertex, 542
exponential function, 52–53
exponential height, 265
exponential search tree, 182, 433
exponential series, 1060
exponentiation
  modular, 879
exponentiation instruction, 22
EXTENDED-EUCLID, 860
EXTEND-SHORTEST-PATHS, 624
extension of a set, 394
exterior of a polygon, 939 ex.
external node, 1088
external path length, 1091 ex.
extracting the maximum key
  from d-ary heaps, 143 pr.
  from max-heaps, 139
extracting the minimum key
  from binomial heaps, 468–469
  from Fibonacci heaps, 482–488
  from 2-3-4 heaps, 473 pr.
  from Young tableaus, 143 pr.
EXTRACT-MAX, 138–139
EXTRACT-MIN, 138, 455
factor, 851
twiddle, 836
factorial function (!), 54–55
factorization, 896–901, 905
  unique, 854
failure in a Bernoulli trial, 1112
fair coin, 1101
fan-out, 988
Farkas’s lemma, 819 pr.
farthest-pair problem, 948
FASTER-ALL-PAIRS-SHORTEST-PATHS, 627, 628 ex.
FASTEST-WAY, 329
Fast Fourier Transform (FFT)
  circuit for, 842–843
  iterative implementation of, 839–842
  multidimensional, 845 pr.
  recursive implementation of, 834–836
using modular arithmetic, 847 pr.
feasibility problem, 601, 818 pr.
feasible linear program, 778
feasible region, 773
feasible solution, 601, 773, 778
Fermat’s theorem, 877
FFT, see Fast Fourier Transform
FIB-HEAP-CHANGE-KEY, 497 pr.
FIB-HEAP-DECREASE-KEY, 489
FIB-HEAP-DELETE, 492
FIB-HEAP-EXTRACT-MIN, 483
FIB-HEAP-INSERT, 480
FIB-HEAP-LINK, 486
FIB-HEAP-PRUNE, 497 pr.
FIB-HEAP-UNION, 482
Fibonacci heap, 476–497
changing a key in, 497 pr.
creating, 480
decreasing a key in, 489–492
deletion from, 492, 496 pr.
in Dijkstra’s algorithm, 599
extracting the minimum key from, 482–488
insertion into, 480–481
in Johnson’s algorithm, 640
maximum degree of, 479, 493–496
minimum key of, 481
potential function for, 479
in Prim’s algorithm, 573
pruning, 497 pr.
running times of operations on, 456 fig.
uniting, 481–482
Fibonacci numbers, 56, 86 pr., 494
computation of, 902 pr.
field of an object, 20
FIFO (first-in, first-out), 200
see also queue
final-state function, 917
FIND-DEPTH, 519 pr.
find path, 506
FIND-SET, 499
disjoint-set-forest implementation of, 508, 522
linked-list implementation of, 501
finished vertex, 540
finishing time, in depth-first search, 541
and strongly connected components, 555
finish time, in activity selection, 371
finite automaton, 916
for string matching, 917–922
FINITE-AUTOMATON-MATCHER, 919
finite group, 862
finite sequence, 1078
finite set, 1073
first-fit heuristic, 1049 pr.
first-in, first-out, 200
see also queue
fixed-length code, 385
floating-point data type, 22
floor function (\lfloor \rfloor), 51
in master theorem, 81–84
floor instruction, 22
flow, 644–650
aggregate, 788
blocking, 697
excess, 669
integer-valued, 666
sum, 650 ex.
total net, 645
total positive, 645
value of, 644
flow conservation, 644
flow network, 644–650
corresponding to a bipartite graph, 665
cut of, 654–657
with negative capacities, 695 pr.
FLOYD-WARSHALL, 630
FLOYD-WARSHALL’, 635 ex.
Floyd-Warshall algorithm, 629–632, 634 ex.
for and loop invariants, 18 n.
in pseudocode, 19
FORD-FULKERSON, 658
Ford-Fulkerson method, 651–664
FORD-FULKERSON-METHOD, 651
forest, 1083, 1085
depth-first, 540
disjoint-set, 505–509
formal power series, 86 pr.
formula-satisfiability problem, 996–998
forward edge, 546
forward substitution, 744–745
fractional knapsack problem, 382, 384 ex.
freeing of objects, 210–212
free list, 211
FREE-OBJECT, 212
free tree, 1083, 1085–1087
frequency domain, 822
full binary tree, 1089, 1091 ex.
relation to optimal code, 386
full node, 439
full rank, 731
full walk of a tree, 1030
fully parenthesized, 331
fully polynomial-time approximation scheme, 1023
for the maximum-clique problem, 1050 pr.
for the subset-sum problem, 1043–1049
function, 1077–1080
Ackermann’s, 521
basis, 762
convex, 1109
linear, 25, 772
objective, see objective function
prefix, 923–925
quadratic, 25
suffix, 917
transition, 916
functional iteration, 55
fundamental theorem of linear programming, 816
fusion tree, 182, 433
fuzzy sorting, 163 pr.
Gabow’s scaling algorithm for single-source shortest paths, 615 pr.
gap character, 910 ex., 923 ex.
gap heuristic, 691 ex.
garbage collection, 127, 210
gate, 987
Gaussian elimination, 747
gcd, 852–853, 856 ex.
see also greatest common divisor
general number-field sieve, 905
generating function, 86 pr.
generator
of a subgroup, 867
of \( \mathbb{Z}_n \), 877
GENERIC-MST, 563
GENERIC-PUSH-RELABEL, 674
generic push-relabel algorithm, 673–681
geometric distribution, 1112
and balls and bins, 109
gometric series, 1060
gift wrapping, 955
global variable, 19
Goldberg’s algorithm, see push-relabel algorithm
golden ratio (\( \phi \)), 56, 86 pr.
good guy, 539 ex.
gossiping, 429
GRAFT, 519 pr.
Graham’s scan, 949–955
GRAHAM-SCAN, 949
graph, 1080–1084
adjacency-list representation of, 528
adjacency-matrix representation of, 529
algorithms for, 525–698
breadth-first search of, 531–539
complement of, 1007
component, 554
constraint, 603–605
dense, 527
depth-first search of, 540–549
dynamic, 499 n.
\( \epsilon \)-dense, 641 pr.
hamiltonian, 979
incidence matrix of, 403 pr., 531 ex.
interval, 379 ex.
nonhamiltonian, 979
shortest path in, 535
singly connected, 549 ex.
sparse, 527
static, 499 n.
tour of, 1012
weighted, 529
see also directed acyclic graph, directed graph, flow network, undirected graph, tree
tree graph-coloring problem, 1019 pr.
graphic matroid, 393, 579
GRAPH-ISOMORPHISM, 982 ex.
grey vertex, 531, 540
greatest common divisor (gcd), 852–853
binary gcd algorithm for, 902 pr.
Euclid’s algorithm for, 856–862
with more than two arguments, 861 ex.
recursion theorem for, 857
greedoid, 404
GREEDY, 396
GREEDY-ACTIVITY-SELECTOR, 378
greedy algorithm, 370–404
for activity selection, 371–379
for coin changing, 402 pr.
compared to dynamic programming, 341, 350 ex., 373–375, 380, 382–383
Dijkstra’s algorithm, 595–601
elements of, 379–384
for fractional knapsack problem, 382
greedy-choice property in, 380–381
for Huffman code, 385–393
Kruskal’s algorithm, 568–570
and matroids, 393–399
for minimum spanning tree, 561
optimal substructure in, 381–382
Prim’s algorithm, 570–573
for the set-covering problem, 1033–1038
for task scheduling, 399–402, 402 pr., 404 pr.
on a weighted matroid, 395–398
for the weighted set-covering problem,1050 pr.
greedy-choice property, 380–381
of Huffman codes, 388–390
of a weighted matroid, 396–397
Greedy-Set-Cover, 1035
grid, 692 pr.
group, 862–869
cyclic, 877
half 3-CNF satisfiability, 1018 ex.
Half-Cleaner, 713
half-open interval, 311
Hall’s theorem, 669 ex.
halving lemma, 832
HAM-CYCLE, 979
hamiltonian cycle, 967, 979
hamiltonian-cycle problem, 979, 1008–1012
hamiltonian graph, 979
hamiltonian path, 983 ex.
hamiltonian-path problem, 1017 ex.
HAM-PATH, 983 ex.
handle, 139, 456, 477
handshaking lemma, 1084 ex.
harmonic number, 1060
harmonic series, 1060
Hash-Delete, 244 ex.
hash function, 224, 229–237
auxiliary, 239
division method for, 230–231
ϵ-universal, 236 ex.
multiplication method for, 231–232
one-way, 886
universal, 232–236
hashing, 221–252
chaining, 225–228, 250 pr.
double, 240–241, 244 ex.
k-universal, 251 pr.
open addressing, 237–245
perfect, 245–249
universal, 232–236
Hash-Insert, 238, 244 ex.
Hash-Search, 238, 244 ex.
hash table, 224–229
dynamic, 424 ex.
secondary, 245
see also hashing
hash value, 224
hat-check problem, 98 ex.
head
in a disk drive, 435
of a linked list, 204
of a queue, 202
heap, 127–144
analyzed by potential method, 416 ex.
binomial, see binomial heap
building, 132–135, 142 pr.
d-ary, 143 pr., 641 pr.
deletion from, 142 ex.
in Dijkstra’s algorithm, 599
extracting the maximum key from, 139
Fibonacci, see Fibonacci heap
as garbage-collected storage, 127
height of, 129
in Huffman’s algorithm, 388
to implement a mergeable heap, 455
increasing a key in, 139–140
insertion into, 140
in Johnson’s algorithm, 640
max-heap, 128
maximum key of, 139
mergeable, see mergeable heap
min-heap, 129
in Prim’s algorithm, 573
as a priority queue, 138–142
relaxed, 497
running times of operations on, 456 fig.
and treaps, 296 pr.
2-3-4, 473 pr.
Heap-Decrease-Key, 141 ex.
Heap-Delete, 142 ex.
Heap-Extract-Max, 139
Heap-Extract-Min, 141 ex.
Heap-Increase-Key, 140
Heap-Maximum, 139
HEAP-MINIMUM, 141 ex.
heap property, 128
  maintenance of, 130–132
  vs. binary-search-tree property, 256 ex.
heapsort, 127–144
HEAPSORT, 136
height
  of a binomial tree, 457
  black, 274
  of a B-tree, 439–440
  of a d-ary heap, 143 pr.
  of a decision tree, 167
  exponential, 265
  of a heap, 129, 129 ex.
  of a node in a heap, 129, 135 ex.
  of a node in a tree, 1088
  of a red-black tree, 274
  of a tree, 1088
height-balanced tree, 296 pr.
height function, in push-relabel algorithms, 671
hereditary family of subsets, 393
Hermitian matrix, 759 ex.
high endpoint of an interval, 311
HIRE-ASSISTANT, 92
hiring problem, 91–92
  on-line, 114–117
  probabilistic analysis of, 97–98
hit, spurious, 912
HOARE-PARTITION, 160 pr.
Hopcroft-Karp bipartite matching algorithm, 696 pr.
horizontal ray, 940 ex.
Horner’s rule, 39 pr., 824
  in the Rabin-Karp algorithm, 911
HUFFMAN, 388
Huffman code, 385–393
hull, convex, 947–957, 964 pr.
hyperedge, 1083
hypergraph, 1083
  and bipartite graphs, 1084 ex.

idempotency laws for sets, 1071
identity, 862
identity matrix, 727
if, in pseudocode, 19
image, 1078
implicit summation notation, 648
inadmissible edge, 681

incidence, 1080
incidence matrix
  and difference constraints, 603
  of a directed graph, 403 pr., 531 ex.
  of an undirected graph, 403 pr.
inclusion and exclusion, 1074 ex.
INCREASE-KEY, 138
increasing a key in a max-heap, 139–140
INCREMENT, 408
incremental design method, 27
  for finding the convex hull, 948
in-degree, 1081
indentation in pseudocode, 19
independence
  of events, 1103, 1106 ex.
  of random variables, 1107
  of subproblems in dynamic programming, 343–344
independent family of subsets, 393
independent set, 1018 pr.
  of tasks, 400
index of an element of \( \mathbb{Z}_n^* \), 877
indicator random variable, 94–98
  in analysis of expected height of a randomly
    built binary search tree, 265–267
  in analysis of inserting into a treap, 296 pr.
  in analysis of streaks, 113–114
  in analysis of the birthday paradox, 108–109
  in approximation algorithm for
    MAX-3-CNF satisfiability, 1040
  in bounding the right tail of the binomial
    distribution, 1122–1123
  in bucket sort analysis, 175–177
  in hashing analysis, 227–228
  in hiring-problem analysis, 97–98
  in quicksort analysis, 157–158, 160 pr.
  in randomized selection analysis, 187–189, 189 ex.
    in universal-hashing analysis, 233–234
induced subgraph, 1082
inequality, linear, 772
inequality constraint, 778
  and equality constraints, 780
infeasible linear program, 778
infeasible solution, 778
infinite sequence, 1078
infinite set, 1073
infinite sum, 1058
infinity, arithmetic with, 587
INDEX

INITIALIZE-PREFLOW, 673
INITIALIZE-SIMPLEX, 796, 812
INITIALIZE-SINGLE-SOURCE, 585
injective function, 1079
inner product, 730
inorder tree walk, 254, 260 ex., 305
INORDER-TREE-WALK, 255
input
to an algorithm, 5
to a combinational circuit, 988
distribution of, 93, 99
to a logic gate, 987
size of, 23
input alphabet, 916
input sequence, 705
input wire, 705
INSERT, 138, 198, 416 ex., 455
insertion
into binary search trees, 261
into binomial heaps, 468
into B-trees, 443–447
into chained hash tables, 226
into d-ary heaps, 143 pr.
into direct-address tables, 222
into dynamic tables, 418
into Fibonacci heaps, 480–481
into heaps, 140
into interval trees, 313
into linked lists, 205–206
into open-address hash tables, 237–238
into order-statistic trees, 306–307
into queues, 201
into red-black trees, 280–287
into stacks, 200
into sweep-line statuses, 942
into 2-3-4 heaps, 473 pr.
into Young tableaus, 143 pr.
insertion sort, 11, 15–19, 24–25
in bucket sort, 174–177
compared to merge sort, 13 ex.
compared to quicksort, 153 ex.
decision tree for, 166 fig.
in merge sort, 37 pr.
in quicksort, 159 ex.
sorting-network implementation of, 708 ex.
using binary search, 37 ex.
INSERTION-SORT, 17, 24
instance
of an abstract problem, 969, 972
of a problem, 5
instructions of the RAM model, 21
integer data type, 22
integer linear-programming problem, 777, 819 pr., 1017 ex.
integers (Z), 1070
integer-valued flow, 666
integral, to approximate summations, 1067
integrality theorem, 667
integration of series, 1060
interior of a polygon, 939 ex.
interior-point method, 776
intermediate vertex, 629
internal node, 1088
internal path length, 1091 ex.
interpolation by a cubic spline, 767 pr.
interpolation by a polynomial, 825, 830 ex.
at complex roots of unity, 836–837
intersection
of chords, 308 ex.
determining, for a set of line segments, 940–947
determining, for two line segments, 936–938
of languages, 976
of sets (∩), 1071
interval, 311
fuzzy sorting of, 163 pr.
INTERVAL-DELETE, 312
interval-graph coloring problem, 379 ex.
INTERVAL-INSERT, 312
INTERVAL-SEARCH, 312, 314
INTERVAL-SEARCH-EXACTLY, 317 ex.
interval tree, 311–317
interval trichotomy, 311
intractability, 966
invalid shift, 906
inverse
of a bijective function, 1079
in a group, 862
of a matrix, 730, 733 ex.
of a matrix from an LUP decomposition, 755–756
multiplicative, modulo n, 871
inversion in a sequence, 39 pr., 99 ex.
inverter, 987
invertible matrix, 730
isolated vertex, 1081
isomorphic graphs, 1082
iterated function, 60 pr.
iterated logarithm function, 55–56
ITERATIVE-FFT, 841
ITERATIVE-TREE-SEARCH, 257

Jarvis’s march, 955
Jensen’s inequality, 1109
JOHNSON, 639
Johnson’s algorithm, 636–640
joining
  of red-black trees, 295 pr.
  of 2-3-4 trees, 453 pr.
joint probability density function, 1107
Josephus permutation, 318 pr.

Karmarkar’s algorithm, 777, 820
Karp’s minimum mean-weight cycle algorithm,
  617 pr.
k-ary tree, 1090
k-CNF, 967
k-coloring, 1019 pr., 1091 pr.
k-combination, 1096
k-conjunctive normal form, 967
kernel of a polygon, 956 ex.
key, 15, 123, 138, 197
  median, of a B-tree node, 443
  public, 881, 884
  secret, 881, 884
  static, 245
Kleene star (*), 976
KMP algorithm, 923–931
KMP-MATCHER, 926
knapack problem
  fractional, 382, 384 ex.
  0–1, 382, 384 ex.
k-neighbor tree, 301
knot, of a spline, 767 pr.
Knuth-Morris-Pratt algorithm, 923–931
k-permutation, 1095
Kraft inequality, 1091 ex.
Kruskal’s algorithm, 568–570
  with integer edge weights, 574 ex.
k-sorted, 180 pr.
k-string, 1095
k-subset, 1073
k-substring, 1095
kth power, 855 ex.
k-universal hashing, 251 pr.
Lagrange’s formula, 826

Lagrange’s theorem, 866
Lamé’s theorem, 859
language, 975
  completeness of, 994 ex.
  proving NP-completeness of, 995–996
  verification of, 980
last-in, first-out, 200
  see also stack
late task, 399
layers
  convex, 962 pr.
  maximal, 962 pr.
LCA, 521 pr.
lcm (least common multiple), 861 ex.
LCS, see longest common subsequence
LCS-LENGTH, 353
leading submatrix, 760
leaf, 1088
least common ancestor, 521 pr.
least common multiple, 861 ex.
least-squares approximation, 762–765
leaving a vertex, 1080
leaving variable, 793
LEFT, 128
left child, 1089
left-child, right-sibling representation, 214,
  217 ex.
LEFT-ROTATE, 278, 316 ex.
left rotation, 277
left spine, 296 pr.
left subtree, 1089
Legendre symbol \((\frac{a}{p})\), 903 pr.
length
  of a path, 1081
  of a sequence, 1078
  of a spine, 296 pr.
  of a string, 907, 1095
level of a function, 510
lexicographically less than, 269 pr.
lexicographic sorting, 269 pr.
\(\lg\) (binary logarithm), 53
\(\lg^*\) (iterated logarithm function), 55–56
\(\lg^k\) (exponentiation of logarithms), 53
\(\lg \lg\) (composition of logarithms), 53
LIFO (last-in, first-out), 200
  see also stack
light edge, 563
linear constraint, 772
linear dependence, 731
Index 1163

linear equality, 772
linear equations
  solving modular, 869–872
  solving systems of, 742–755
  solving tridiagonal systems of, 767 pr.
linear function, 25, 772
linear independence, 731
linear inequality, 772
linear-inequality feasibility problem, 818 pr.
  linearity of expectation, 1108
  linearity of summations, 1059
  linear order, 1077
linear probing, 239
linear programming, 770–821
  algorithms for, 776–777
  applications of, 776
  duality in, 804–811
  finding an initial solution for, 811–816
  fundamental theorem of, 816
  interior-point methods for, 776, 820
  Karmarkar’s algorithm for, 777, 820
  and maximum flow, 786
  and minimum-cost flow, 787–788
  and minimum-cost multicommodity flow, 790 ex.
  and multicommodity flow, 788–789
  simplex algorithm for, 790–804
  and single-pair shortest path, 785–786
  and single-source shortest paths, 601–607
  slack form for, 781–783
  standard form for, 777–781
  see also integer linear-programming problem, 0-1 integer-programming problem
linear-programming relaxation, 1041
linear search, 21 ex.
line segment, 934
  determining turn of, 936
  determining whether any intersect, 940–947
  determining whether two intersect, 936–938
link
  of binomial trees, 457
  of Fibonacci-heap roots, 483
LINK, 508
linked list, 204–209
  compact, 213 ex., 218 pr.
  deletion from, 206
  to implement disjoint sets, 501–505
  insertion into, 205–206
neighbor list, 683
searching, 205, 236 ex.
list, see linked list
LIST-DELETE, 206
LIST-DELETE’, 206
LIST-INSERT, 206
LIST-INSERT’, 207
LIST-SEARCH, 205
LIST-SEARCH’, 207
literal, 999
little-oh notation, 47–48
little-omega notation, 48
L∞-distance, 962 ex.
ln (natural logarithm), 53
load factor
  of a dynamic table, 417
  of a hash table, 226
load instruction, 22
local variable, 19
logarithm function (log), 53–54
  discrete, 877
  iterated (lgk), 55–56
logic gate, 987
longest common subsequence, 350–356, 369
LONGEST-PATH, 978 ex.
LONGEST-PATH-LENGTH, 978 ex.
longest-simple-cycle problem, 1017 ex.
longest simple path, 966
  in an unweighted graph, 342
LOOKUP-CHAIN, 348
looping constructs in pseudocode, 19
loop invariant, 17
  for breadth-first search, 534
  for building a heap, 133
  for consolidating the root list in extracting the minimum node from a Fibonacci heap, 486
  for determining the rank of an element in an order-statistic tree, 305
  for Dijkstra’s algorithm, 597
  for the generic minimum-spanning-tree algorithm, 562
  for the generic push-relabel algorithm, 676
  for heapsort, 136 ex.
  for Horner’s rule, 39 pr.
  for increasing a key in a heap, 142 ex.
  initialization of, 18
  for insertion sort, 17–19
  and for loops, 18 n.
maintenance of, 18
for merging, 30
for modular exponentiation, 879–880
origin of, 40
for partitioning, 146
for Prim’s algorithm, 572
for the Rabin-Karp algorithm, 914
for randomly permuting an array, 103
for red-black tree insertion, 283
for the relabel-to-front algorithm, 687
for searching an interval tree, 315
for simplex algorithm, 798
for string-matching automata, 919, 921
and termination, 18
for uniting binomial heaps, 472 ex.
low endpoint of an interval, 311
lower bounds
for average sorting, 180 pr.
for convex hull, 956 ex.
for median finding, 195
for merging, 180 pr.
for minimum-weight vertex cover, 1042
and potential functions, 429
for size of a merging network, 718 ex.
for size of optimal vertex cover, 1026
for sorting, 165–168
lower median, 183
lower-triangular matrix, 728
LU decomposition, 747–750
LU-DECOMPOSITION, 749
LUP decomposition, 743
computation of, 750–754
of a diagonal matrix, 754 ex.
in matrix inversion, 755–756
and matrix multiplication, 759 ex.
of a permutation matrix, 754 ex.
use of, 743–747
LUP-DECOMPOSITION, 752
LUP-SOLVE, 745
main memory, 434
MAKE-BINOMIAL-HEAP, 461
MAKE-HEAP, 455
MAKE-SET, 498
disjoint-set-forest implementation of, 508
linked-list implementation of, 501
MAKE-TREE, 519 pr.
Manhattan distance, 194 pr., 962 ex.
marked node, 478, 490
Markov’s inequality, 1111 ex.
master method for solving a recurrence, 73–76
master theorem, 73
proof of, 76–84
matched vertex, 664
matching
maximal, 1026, 1051 pr.
maximum, 1051 pr.
and maximum flow, 664–669
of strings, 906–932
weighted bipartite, 497
matroid, 393
matrix, 725–734
adjacency, 529
conjugate transpose of, 759 ex.
Hermitian, 759 ex.
incidence, 403 pr., 531 ex.
pseudoinverse of, 764
symmetric positive-definite, 760–762
Toeplitz, 844 pr.
transpose of, 529, 726
see also matrix inversion, matrix multiplication
matrix-chain multiplication, 331–339
MATRIX-CHAIN-MULTIPLY
MATRIX-CHAIN-ORDER, 336
matrix inversion, 755–758
matrix multiplication
for all-pairs shortest paths, 622–629
boolean, 759 ex.
and computing the determinant, 759 ex.
and LUP decomposition, 759 ex.
and matrix inversion, 756–758
Pan’s method for, 741 ex.
Strassen’s algorithm for, 735–742
MATRIX-MULTIPLY, 332, 625
matroid, 393–399, 403 pr., 579
MAX-CNF satisfiability, 1043 ex.
MAX-CUT problem, 1043 ex.
MAX-FLOW-BY-SCALING, 694 pr.
max-flow min-cut theorem, 657
max-heap, 128
building, 132–135
deletion from, 142 ex.
extracting the maximum key from, 139
in heapsort, 135–138
increasing a key in, 139–140
insertion into, 140
maximum key of, 139
as a max-priority queue, 138–142
mergeable, see mergeable max-heap
MAX-HEAPIFY, 130
MAX-HEAP-INSERT, 140
building a heap with, 142 pr.
max-heap property, 128
maintenance of, 130–132
maximal element of a partially ordered set, 1076
maximal layers, 962 pr.
maximal matching, 1026, 1051 pr.
maximal point, 962 pr.
maximal subset in a matroid, 394
maximization linear program, 773
and minimization linear programs, 779
maximum, 183
in binary search trees, 258
of a binomial distribution, 1117 ex.
finding, 184–185
in heaps, 139
in order-statistic trees, 310 ex.
in red-black trees, 276
MAXIMUM, 138–139, 198
maximum bipartite matching, 664–669, 680 ex.
Hopcroft-Karp algorithm for, 696 pr.
maximum degree in a Fibonacci heap, 479, 488 ex., 493–496
maximum flow, 643–698
Edmonds-Karp algorithm for, 660–663
Ford-Fulkerson method for, 651–664
as a linear program, 786
and maximum bipartite matching, 664–669
with negative capacities, 695 pr.
push-relabel algorithms for, 669–692
relabel-to-front algorithm for, 681–692
scaling algorithm for, 694 pr.
updating, 694 pr.
maximum matching, 1051 pr.
max-priority queue, 138
MAX-3-CNF satisfiability, 1039–1040
MAYBE-MST-A, 578 pr.
MAYBE-MST-B, 578 pr.
MAYBE-MST-C, 578 pr.
mean, see expected value
mean weight of a cycle, 617 pr.
median, 183–195
of sorted lists, 193 ex.
weighted, 194 pr.
median key of a B-tree node, 443
median-of-3 method, 162 pr.
member of a set (\(\in\)), 1070
memoization, 347–349
MEMOIZED-MATRIX-CHAIN, 348
memory, 434
memory hierarchy, 22
merge
of \(k\) sorted lists, 142 ex.
lower bounds for, 180 pr.
of two sorted arrays, 28
using a comparison network, 716–718
MERGE, 29
mergeable heap, 431, 455
and comparison sorts, 489 ex.
linked-list implementation of, 217 pr.
relaxed heaps, 497
running times of operations on, 456 fig.
2-3-4 heaps, 473 pr.
see also binomial heap, Fibonacci heap
mergeable max-heap, 217 n., 431 n., 455 n.
mergeable min-heap, 217 n., 431 n., 455
MERGE-LISTS, 1044
MERGER, 717
merge sort, 11, 28–36
compared to insertion sort, 13 ex.
sorting-network implementation of, 719–721
use of insertion sort in, 37 pr.
MERGE-SORT, 32
recursion tree for, 349 ex.
merging network, 716–718
odd-even, 721 pr.
MILLER-RABIN, 892
Miller-Rabin primality test, 890–896
MIN-GAP, 317 ex.
min-heap, 129
analyzed by potential method, 416 ex.
building, 132–135
in Dijkstra’s algorithm, 599
in Huffman’s algorithm, 388
in Johnson’s algorithm, 640
mergeable, see mergeable min-heap
as a min-priority queue, 141 ex.
in Prim’s algorithm, 573
MIN-HEAPIFY, 132 ex.
MIN-HEAP-INSERT, 141 ex.
min-heap-ordered, 459
min-heap property, 129, 459
maintenance of, 132 ex.
vs. binary-search-tree property, 256 ex.
minimization linear program, 773
and maximization linear programs, 779
minimum, 183
in binary search trees, 258
in binomial heaps, 462
in B-trees, 447 ex.
in Fibonacci heaps, 481
finding, 184–185
off-line, 518 pr.
in order-statistic trees, 310 ex.
in red-black trees, 276
Minimum, 138, 184, 198, 455
minimum-cost flow, 787–788
minimum-cost multicommodity flow, 790 ex.
minimum-cost spanning tree, see minimum spanning tree
minimum cut, 655
minimum degree of a B-tree, 439
minimum key in 2-3-4 heaps, 473 pr.
minimum mean-weight cycle, 617 pr.
minimum node of a Fibonacci heap, 478
minimum path cover, 692 pr.
minimum spanning tree, 561–579
in approximation algorithm for
traveling-salesman problem, 1028
Borůvka’s algorithm for, 578
constructed using binomial heaps, 474 pr.
on dynamic graphs, 574 ex.
generic algorithm for, 562–567
Kruskal’s algorithm for, 568–570
Prim’s algorithm for, 570–573
relation to matroids, 393, 395
second-best, 575 pr.
minimum-weight spanning tree, see minimum spanning tree
minimum-weight vertex cover, 1040–1043
minor of a matrix, 732
min-priority queue, 138, 141 ex.
in constructing Huffman codes, 387
in Dijkstra’s algorithm, 598
in Prim’s algorithm, 572–573
mirroring, 833 n.
missing child, 1089
mod, 51, 851
modular arithmetic, 51–52, 862–869
modular exponentiation, 879
Modular-exponentiation, 879
modular linear equations, 869–872
Modular-linear-equation-solver, 871
modulo, 51, 851
Monge array, 88 pr.
monotone sequence, 144
monotonically decreasing, 51
monotonically increasing, 51
MST, 474 pr.
Mst-Kruskal, 569
Mst-Prim, 572
Mst-reduce, 576 pr.
multicommodity flow, 788–789
minimum-cost, 790 ex.
multidimensional Fast Fourier Transform, 845 pr.
multigraph, 1083
converting to equivalent undirected graph, 530 ex.
multiple, 729, 850
of an element modulo n, 869–872
least common, 861 ex.
multiple assignment, 19
multiple sources and sinks, 647
multiplication
of complex numbers, 741 ex.
divide-and-conquer method for, 844 pr.
of matrices, 729, 734 ex.
of a matrix chain, see matrix-chain multiplication
modulo n \( (\cdot n) \), 863
of polynomials, 823
multiplication method, 231–232
multiplicative group modulo n, 864
multiplicative inverse, modulo n, 871
multiply instruction, 22
Multipop, 406
Multipush, 409 ex.
multiset, 1070 n.
mutually exclusive events, 1100
mutually independent events, 1103
N (set of natural numbers), 1070
naive algorithm for string matching, 909–911
Naive-string-matcher, 909
natural cubic spline, 767 pr.
natural numbers (N), 1070
negative of a matrix, 729
negative-weight cycle
and difference constraints, 603
and relaxation, 613 ex.
and shortest paths, 582
negative-weight edges, 582–583
neighbor, 1083
neighborhood, 669 ex.
neighbor list, 683
nesting boxes, 615 pr.
net
flow across a cut, 655
network
admissible, 681–683
bitonic sorting, 712–716
comparison, 704–709
flow, see flow network
for merging, 716–718
odd-even merging, 721 pr.
odd-even sorting, 721 pr.
permutation, 722 pr.
residual, 651–653
sorting, 704–724
transposition, 721 pr.
NEXT-TO-TOP, 949
NIL, 20
node, 1087
see also vertex
nonbasic variable, 782
nondeterministic polynomial time, 981 n.
see also NP
nonhamiltonian graph, 979
noninstance, 974 n.
noninvertible matrix, 730
nonnegativity constraint, 777, 779
nonoverlappable string pattern, 922 ex.
nonsaturating push, 672, 678
nonsingular matrix, 730
nontrivial power, 855 ex.
nontrivial square root of 1, modulo \( n \), 878
no-path property, 587, 608–609
normal equation, 764
norm of a vector, 730
NOT function (\( \neg \)), 987
NOT gate, 987
NP (complexity class), 967, 981, 983 ex.
NPC (complexity class), 968, 986
NP-complete, 968, 986
NP-completeness, 966–1021
of the circuit-satisfiability problem, 987–994
of the clique problem, 1003–1006
of determining whether a boolean formula is
a tautology, 1002 ex.
of the formula-satisfiability problem,
996–998
of the graph-coloring problem, 1019 pr.
of the half 3-CNF satisfiability problem,
1018 ex.
of the hamiltonian-cycle problem,
1008–1012
of the hamiltonian-path problem, 1017 ex.
of the independent-set problem, 1018 pr.
of the integer linear-programming problem,
1017 ex.
of the longest-simple-cycle problem,
1017 ex.
proving, of a language, 995–996
of scheduling with profits and deadlines,
1020 pr.
of the set-covering problem, 1038 ex.
of the set-partition problem, 1017 ex.
of the subgraph-isomorphism problem,
1017 ex.
of the subset-sum problem, 1013–1017
of the 3-CNF-satisfiability problem,
998–1002
of the traveling-salesman problem,
1012–1013
of the vertex-cover problem, 1006–1008
of the 0-1 integer-programming problem,
1017 ex.
NP-hard, 986
n-set, 1073
n-tuple, 1074
null event, 1100
null tree, 1089
null vector, 731
number-field sieve, 905
numerical stability, 725, 743, 769
o-notation, 47–48
O-notation, 43 fig., 44–45
\( O' \)-notation, 59 pr.
\( \tilde{O} \)-notation, 59 pr.
object, 20
allocation and freeing of, 210–212
array implementation of, 209–213
passing as parameter, 20
objective function, 601, 606–607 ex., 773, 777
objective value, 774, 778
optimal, 778
occurrence of a pattern, 906
odd-even merging network, 721 pr.
odd-even sorting network, 721 pr.
Off-Line-Minimum, 519 pr.
off-line problem
  least common ancestors, 521 pr.
  minimum, 518 pr.
Omega-notation, 43 fig., 45–46
1-approximation algorithm, 1023
one-pass method, 522
one-to-one correspondence, 1079
one-to-one function, 1079
one-way hash function, 886
on-line convex-hull problem, 957 ex.
on-line hiring problem, 114–117
On-Line-Maximum, 115
On-Segment, 937
onto, 1079
open-address hash table, 237–245
  double hashing, 240–241, 244 ex.
  linear probing, 239
  quadratic probing, 239–240, 250 pr.
on interval, 311
optimal binary search tree, 356–363, 369
Optimal-BST, 361
optimal objective value, 778
optimal solution, 778
optimal subset of a matroid, 395
optimal substructure
  of activity selection, 371–373
  of assembly-line scheduling, 325–327
  of binary search trees, 359
  in dynamic programming, 339–344
  of the fractional knapsack problem, 382
  in greedy algorithms, 381–382
  of Huffman codes, 391
  of longest common subsequences, 351–352
  of matrix-chain multiplication, 333–334
  of shortest paths, 581–582, 623, 629
  of unweighted shortest paths, 342
  of weighted matroids, 397
  of the 0-1 knapsack problem, 382
optimal vertex cover, 1024
optimization problem, 323, 968, 972
  approximation algorithms for, 1022–1054
  and decision problems, 969
OR function (\lor), 633, 987
or, in pseudocode, 20
order
  of a group, 867
  linear, 1077
  partial, 1076
  total, 1077
ordered pair, 1073
ordered tree, 1088
order of growth, 26
order statistics, 183–195
  dynamic, 302–308
order-statistic tree, 302–308
  querying, 310 ex.
OR gate, 987
origin, 934
orthonormal, 769
OS-Key-Rank, 307 ex.
OS-Rank, 305
OS-Select, 304
out-degree, 1081
outer product, 730
output
  of an algorithm, 5
  of a combinational circuit, 988
  of a logic gate, 987
output sequence, 705
output wire, 705
overdetermined system of linear equations, 743
overflow
  of a queue, 202
  of a stack, 201
overflowing vertex, 670
overlapping intervals, 311
  finding all, 317 ex.
  point of maximum overlap, 318 pr.
overlapping rectangles, 317 ex.
overlapping subproblems, 344–346
overlapping-suffix lemma, 908
P (complexity class), 967, 973, 977, 979 ex.
package wrapping, 955
page on a disk, 436, 452 pr.
paging, 22
pair, ordered, 1073
pairwise disjoint sets, 1073
pairwise independence, 1103
pairwise relatively prime, 854
Pan’s method for matrix multiplication, 741 ex.
parallel-machine-scheduling problem, 1051 pr.
parameter, 20
costs of passing, 85 pr.
parent, 1087
  in a breadth-first tree, 532
PARENT, 128
parenthesis structure of depth-first search, 543
costs of passing, 85 pr.
parenthesis theorem, 543
costs of passing, 85 pr.
parenthesization of a matrix-chain product, 331
costs of passing, 85 pr.
parsing, 999
costs of passing, 85 pr.
partial order, 1076
PARTITION, 146
costs of passing, 85 pr.
partitioning algorithm, 146–148
costs of passing, 85 pr.
  around median of 3 elements, 159 ex.
costs of passing, 85 pr.
randomized, 154
partition of a set, 1073, 1076
Pascal’s triangle, 1099 ex.
costs of passing, 85 pr.
path, 1081
  augmenting, 654, 696 pr.
costs of passing, 85 pr.
  critical, 594
costs of passing, 85 pr.
  find, 506
  hamiltonian, 983 ex.
costs of passing, 85 pr.
  longest, 342, 966
  shortest, see shortest paths
PATH, 969, 976
  weight of, 580
path compression, 506
PATH, 969, 976
  path length, of a tree, 270 pr., 1091 ex.
costs of passing, 85 pr.
path-relaxation property, 587, 609–610
PATH, 969, 976
  pattern in string matching, 906
costs of passing, 85 pr.
  nonoverlapping, 922 ex.
costs of passing, 85 pr.
pattern matching, see string matching
  penalty, 399
costs of passing, 85 pr.
  perfect hashing, 245–249
  perfect matching, 669 ex.
costs of passing, 85 pr.
  permutation, 1079
  bit-reversal, 425 pr., 841
  Josephus, 318 pr.
  in place, 102
  random, 101–104
  of a set, 1095
  uniform random, 93, 101
costs of passing, 85 pr.
  permutation matrix, 728, 733 ex.
  LUP decomposition of, 754 ex.
costs of passing, 85 pr.
  permutation network, 722 pr.
PERMUTE-BY-CYCLIC, 105 ex.
costs of passing, 85 pr.
PERMUTE-BY-SORTING, 101
PERMUTE-WITH-ALL, 105 ex.
costs of passing, 85 pr.
PERMUTE-WITHOUT-IDENTITY, 104 ex.
costs of passing, 85 pr.
persistent data structure, 294 pr., 432
PERSISTENT-TREE-INSERT, 294 pr.
PERSISTENT-TREE-INSERT, 294 pr.
PERSISTENT-TREE-INSERT, 294 pr.
PERSISTENT-TREE-INSERT, 294 pr.
PERSISTENT-TREE-INSERT, 294 pr.
PERSISTENT-TREE-INSERT, 294 pr.
PERSISTENT-TREE-INSERT, 294 pr.
PERSISTENT-TREE-INSERT, 294 pr.
PERSISTENT-TREE-INSERT, 294 pr.
PERSISTENT-TREE-INSERT, 294 pr.
PERSISTENT-TREE-INSERT, 294 pr.
PERSISTENT-TREE-INSERT, 294 pr.
PERSISTENT-TREE-INSERT, 294 pr.
PERSISTENT-TREE-INSERT, 294 pr.
PERSISTENT-TREE-INSERT, 294 pr.
PERSISTENT-TREE-INSERT, 294 pr.
PERSISTENT-TREE-INSERT, 294 pr.
PERSISTENT-TREE-INSERT, 294 pr.
PERSISTENT-TREE-INSERT, 294 pr.
for lower bounds, 429
potential method, 412–416
for binary counters, 414–415
for disjoint-set data structures, 512–517
for dynamic tables, 419–420, 422–424
for Fibonacci heaps, 479–482, 487–488, 491–492
for the generic push-relabel algorithm, 678–679
for the Knuth-Morris-Pratt algorithm, 926–927
for min-heaps, 416 ex.
for restructuring red-black trees, 428 pr.
for stack operations, 413–414
for the Knuth-Morris-Pratt algorithm, 926–927
for min-heaps, 416 ex.
for restructuring red-black trees, 428 pr.
for stack operations, 413–414
potential of a data structure, 413
power
of an element, modulo \( n \), 876–881
\( k \)th, 855 ex.
nontrivial, 855 ex.
power series, 86 pr.
power set, 1073
Pr \{ \} (probability distribution), 1100
predecessor
in binary search trees, 258–259
in breadth-first trees, 532
in B-trees, 447 ex.
in linked lists, 204
in order-statistic trees, 310 ex.
in red-black trees, 276
in shortest-paths trees, 584
Predecessor, 198
predecessor matrix, 621
predecessor subgraph
in all-pairs shortest paths, 621
in breadth-first search, 537
in depth-first search, 540
in single-source shortest paths, 584
predecessor-subgraph property, 587, 612–613
preemption, 402 pr.
prefix
of a sequence, 351
of a string (□), 907
prefix code, 385
prefix function, 923–925
prefix-function iteration lemma, 927
prefixflow, 669
preorder tree walk, 254
presorting, 961
Prim’s algorithm, 570–573
with an adjacency matrix, 573 ex.
in approximation algorithm for
traveling-salesman problem, 1028
implemented with a Fibonacci heap, 573
implemented with a min-heap, 573
with integer edge weights, 574 ex.
similarity to Dijkstra’s algorithm, 570, 599
for sparse graphs, 575 pr.
primality testing, 887–896, 904
Miller-Rabin test, 890–896
pseudoprimality testing, 889–890
primal linear program, 805
primary clustering, 239
primary memory, 434
prime distribution function, 888
prime number, 851
density of, 887–888
prime number theorem, 888
primitive root of \( \mathbb{Z}_n^\ast \), 877
principal root of unity, 831
principle of inclusion and exclusion, 1074 ex.
Print-All-Pairs-Shortest-Path, 621
Print-Intersecting-Segments, 946 ex.
Print-LCS, 355
Print-Optimal-Parens, 338, 338 ex.
Print-Path, 538
Print-Stations, 330
priority queue, 138–142
in constructing Huffman codes, 387
in Dijkstra’s algorithm, 598
heap implementation of, 138–142
max-priority queue, 138
min-priority queue, 138, 141 ex.
with monotone extractions, 144
in Prim’s algorithm, 572–573
see also binary search tree, binomial heap,
Fibonacci heap
probabilistic analysis, 92–93, 106–117
of approximation algorithm for
MAX-3-CNF satisfiability, 1040
of average-case lower bound for sorting, 178 pr.
and average inputs, 26
of average node depth in a randomly built
binary search tree, 270 pr.
of balls and bins, 109–110
of birthday paradox, 106–109
of bucket sort, 174–177, 177 ex.
of collisions, 228 ex., 249 ex.
of convex hull over a sparse-hulled distribution, 964 pr.
of file comparison, 915 ex.
of hashing with chaining, 226–228
of the height of a randomly built binary search tree, 265–268
of the hiring problem, 97–98
of insertion into a binary search tree with equal keys, 268 pr.
of longest-probe bound for hashing, 249 pr.
of the Miller-Rabin primality test, 893–896
of open-address hashing, 241–244, 244 ex.
of partitioning, 153 ex., 159 ex., 160 pr., 162 pr.
of perfect hashing, 246–249
of Pollard’s rho heuristic, 898–901
of probabilistic counting, 118 pr.
of quicksort, 156–160, 162 pr., 268 ex.
of the Rabin-Karp algorithm, 915
and randomized algorithms, 99–101
of searching a compact list, 218 pr.
of slot-size bound for chaining, 250 pr.
of sorting points by distance from origin, 177 ex.
of streaks, 110–114
of universal hashing, 233–236
probabilistic counting, 118 pr.
probability, 1100–1106
probability density function, 1107
probability distribution, 1100
probability distribution function, 177 ex.
probe, 237, 249 pr.
probe sequence, 237
probing, see linear probing, quadratic probing
problem
abstract, 972
computational, 5–6
concrete, 973
decision, 969, 972
intractable, 966
optimization, 323, 968, 972
solution to, 6, 972–973
tractable, 966
procedure, 6, 15–16
product
Cartesian, 1074
cross, 934
priority, see priority queue
in push-relabel algorithms, 691 ex.
quick sort, 145–164
analysis of, 149–153, 155–159
average-case analysis of, 156–158
compared to insertion sort, 153 ex.
compared to radix sort, 173
description of, 145–149
good worst-case implementation of, 192 ex.
with median-of-3 method, 162 pr.
randomized version of, 153–154, 160 pr.
stack depth of, 162 pr.
tail-recursive version of, 162 pr.
use of insertion sort in, 159 ex.
worst-case analysis of, 155
QUICKSORT, 146
QUICKSORT’, 162 pr.
quotient, 851
R (set of real numbers), 1070
Rabin-Karp algorithm, 911–916
RABIN-KARP-MATCHER, 914
radix sort, 170–173
compared to quick sort, 173
RADIX-SORT, 172
radix tree, 269 pr.
RAM, see random-access machine
RANDOM, 94, 94 ex.
random-access machine, 21–22
vs. comparison networks, 704
randomized algorithm, 93–94, 99–105
and average inputs, 26
comparison sort, 178 pr.
for the hiring problem, 100
for insertion into a binary search tree with
equal keys, 268 pr.
for MAX-3-CNF satisfiability, 1039–1040
Miller-Rabin primality test, 890–896
for partitioning, 154, 159 ex., 160 pr., 162 pr.
for permuting an array, 101–104
Pollard’s rho heuristic, 897–901, 901 ex.
and probabilistic analysis, 99–101
quick sort, 153–154, 159 ex., 160 pr., 162 pr.
randomized rounding, 1053
for searching a compact list, 218 pr.
for selection, 185–189
universal hashing, 232–236
worst-case performance of, 154 ex.
RANDOMIZED-HIRE-ASSISTANT, 100
RANDOMIZED-PARTITION, 154
RANDOMIZED-QUICKSORT, 154, 268 ex.
relation to randomly built binary search
trees, 270 pr.
randomized rounding, 1053
RANDOMIZED-SELECT, 186
RANDOMIZE-IN-PLACE, 103
randomly built binary search tree, 265–268,
270 pr.
random-number generator, 94
random permutation, 101–104
uniform, 93, 101
random sampling, 154
RANDOM-SEARCH, 118 pr.
random variable, 1106–1111
indicator, see indicator random variable
range, 1078
rank
column, 731
full, 731
of a matrix, 731, 734 ex.
of a node in a disjoint-set forest, 506,
511–512, 518 ex.
of a number in an ordered set, 302
row, 731
rate of growth, 26
ray, 940 ex.
RB-DELETE, 288
RB-DELETE-FIXUP, 289
RB-ENUMERATE, 311 ex.
RB-INSERT, 280
RB-INSERT-FIXUP, 281
RB-JOIN, 295 pr.
reachability in a graph (→), 1081
real numbers (R), 1070
reconstructing an optimal solution in dynamic
programming, 346–347
record (data), 123
rectangle, 317 ex.
recurrence, 32, 62–90
solution by Akra-Bazzi method, 89–90
solution by master method, 73–76
solution by recursion-tree method, 67–72
solution by substitution method, 63–67
recurrence equation, see recurrence
recursion, 28
recursion tree, 36, 67–72
for merge sort, 349 ex.
Index

in proof of master theorem, 76–78
and the substitution method, 70–72
RECURSIVE-ACTIVITY-SELECTOR, 376
RECURSIVE-FFT, 835
RECURSIVE-MATRIX-CHAIN, 345
red-black tree, 273–301
  augmentation of, 309–310
  compared to B-trees, 440
  deletion from, 288–294
  in determining whether any line segments intersect, 943
  for enumerating keys in a range, 311 ex.
  height of, 274
  insertion into, 280–287
  joining of, 295 pr.
  maximum key of, 276
  minimum key of, 276
  predecessor in, 276
  properties of, 273–277
  relaxed, 276 ex.
  restructuring, 428 pr.
  rotation in, 277–279
  searching in, 276
  successor in, 276
  and 2-3-4 trees, 441 ex.
see also trees, order-statistic tree
reducibility, 984–986
reduction algorithm, 970, 984
reduction function, 984
reflective relation, 1075
reflectivity of asymptotic notation, 49
region, feasible, 773
rejection
  by an algorithm, 976
  by a finite automaton, 917
RELABEL, 673
relabeled vertex, 673
relabel operation (in push-relabel algorithms),
  672–673, 677
RELABEL-TO-FRONT, 687
relabel-to-front algorithm, 681–692
relation, 1075–1077
relatively prime, 854
RELAX, 586
relaxation
  of an edge, 585–587
  linear programming, 1041
relaxed heap, 497
relaxed red-black tree, 276 ex.
release time, 402 pr.
remainder, 51, 851
remainder instruction, 22
repeat, in pseudocode, 19
repeated squaring
  for all-pairs shortest paths, 625–627
  for raising a number to a power, 879
repetition factor of a string, 931 pr.
REPETITION-MATCHER, 931 pr.
representative of a set, 498
RESET, 412 ex.
residual capacity, 651, 654
residual edge, 652
residual network, 651–653
residue, 51, 851, 903 pr.
respect a set of edges, 563
return instruction, 22
reweighting
  in all-pairs shortest paths, 636
  in single-source shortest paths, 615 pr.
rho heuristic, 897–901, 901 ex.
$\rho(n)$-approximation algorithm, 1022
RIGHT, 128
right child, 1089
right-conver, 279 ex.
right horizontal ray, 940 ex.
RIGHT-ROTATE, 278
right rotation, 277
right spine, 296 pr.
right subtree, 1089
root
  of a tree, 1087
  of unity, 830–831
  of $\mathbb{Z}_n^n$, 877
rooted tree, 1087
representation of, 214–217
root list
  of a binomial heap, 459
  of a Fibonacci heap, 478
rotation
  cyclic, 930 ex.
  in a red-black tree, 277–279
rotational sweep, 947, 949–955
rounding, 1042
  randomized, 1053
row rank, 731
row vector, 726
RSA public-key cryptosystem, 881–887
rule of product, 1095
rule of sum, 1094
running time, 23
  average-case, 26
  best-case, 27 ex., 46
  of a comparison network, 707
  expected, 26
  of a graph algorithm, 526
  order of growth, 26
  rate of growth, 26
  worst-case, 26, 46

safe edge, 562
SAME-COMPONENT, 500
sample space, 1100
sampling, 154
SAT, 996
satellite data, 123, 197
satisfiability, 988, 996–998, 1039–1040, 1043 ex.
satisfiable formula, 967, 996
satisfying assignment, 988, 996
saturated edge, 672
saturating push, 672, 678
scalar flow product, 650 ex.
scalar multiple, 729
scaling
  in maximum flow, 694 pr.
  in single-source shortest paths, 615 pr.
scapegoat tree, 301
schedule, 399, 1051 pr.
  event-point, 942
scheduling, 369 pr., 402 pr., 1020 pr., 1051 pr.
Schur complement, 748, 761
Schur complement lemma, 761
SCRAMBLE-SEARCH, 118 pr.
SEARCH, 198
searching
  binary search, 37 ex.
  in binary search trees, 256–258
  in B-trees, 441–442
  in chained hash tables, 226
  in compact lists, 218 pr.
  in direct-address tables, 222
  for an exact interval, 317 ex.
  in interval trees, 314–316
  linear search, 21 ex.
  in linked lists, 205
  in open-address hash tables, 238
problem of, 21 ex.
  in red-black trees, 276
  an unsorted array, 118 pr.
search tree, see balanced search tree, binary
search tree, B-tree, exponential search
tree, interval tree, optimal binary search
tree, order-statistic tree, red-black tree,
splay tree, 2-3 tree, 2-3-4 tree
secondary clustering, 240
secondary hash table, 245
secondary storage
  search tree for, 434–454
  stacks on, 452 pr.
second-best minimum spanning tree, 575 pr.
secret key, 881, 884
segment, see directed segment, line segment
SEGMENTS-INTERSECT, 937
SELECT, 189–190
selection
  of activities, see activity-selection problem
  and comparison sorts, 192
  in expected linear time, 185–189
  in order-statistic trees, 303–304
  problem of, 183
  in worst-case linear time, 189–193
selection sort, 27 ex.
selector vertex, 1009
self-loop, 1080
semicomponented graph, 557 ex.
sentinel, 29, 206–208, 274
sequence (⟨⟩)
  bitonic, 618 pr., 712
  clean, 713
  finite, 1078
  infinite, 1078
  input, 705
  inversion in, 39 pr., 99 ex.
  output, 705
  probe, 237
series, 86 pr., 1059–1061
set (⟨⟩), 1070–1075
  convex, 650 ex.
  independent, 1018 pr.
set-covering problem, 1033–1038
  weighted, 1050 pr.
set-partition problem, 1017 ex.
shadow of a point, 956 ex.
Shell’s sort, 40
shift in string matching, 906
shift instruction, 22
Index

short-circuiting operator, 20
SHORTEST-PATH, 968
shortest paths, 580–642
  all-pairs, 581, 620–642
  Bellman-Ford algorithm for, 588–592
  with bitonic paths, 618 pr.
  and breadth-first search, 534–537, 581
  convergence property of, 587, 609
  and difference constraints, 601–607
  Dijkstra’s algorithm for, 595–601
  in a directed acyclic graph, 592–595
  in ϵ-dense graphs, 641 pr.
  estimate of, 585
  Floyd-Warshall algorithm for, 629–632
  Gabow’s scaling algorithm for, 615 pr.
  Johnson’s algorithm for, 636–640
  as a linear program, 785–786
  and longest paths, 966
  by matrix multiplication, 622–629
  and negative-weight cycles, 582
  with negative-weight edges, 582–583
  no-path property of, 587, 608–609
  optimal substructure of, 581–582
  path-relaxation property of, 587, 609–610
  predecessor-subgraph property of, 587, 612–613
  problem variants, 581
  and relaxation, 585–587
  by repeated squaring, 625–627
  single-destination, 581
  single-pair, 341, 581
  single-source, 580–619
  tree of, 584, 610–613
  triangle inequality of, 587, 607–608
  in an unweighted graph, 341, 535
  upper-bound property of, 587, 608
  in a weighted graph, 580
sibling, 1088
side of a polygon, 939 ex.
signature, 883
simple cycle, 1081
simple graph, 1082
simple path, 1081
  longest, 342, 966
simple polygon, 939 ex.
simple uniform hashing, 226
simplex, 775
SIMPLEX, 797
simplex algorithm, 775, 790–804, 820–821
single-destination shortest paths, 581
single-pair shortest path, 341, 581
  as a linear program, 785–786
single-source shortest paths, 580–619
  Bellman-Ford algorithm for, 588–592
  with bitonic paths, 618 pr.
  and difference constraints, 601–607
  Dijkstra’s algorithm for, 595–601
  in a directed acyclic graph, 592–595
  in ϵ-dense graphs, 641 pr.
  Gabow’s scaling algorithm for, 615 pr.
  and longest paths, 966
singleton, 1073
singly connected graph, 549 ex.
singly linked list, 204
see also linked list
singular matrix, 730
singular value decomposition, 769
sink, 530 ex., 644, 647
size
  of an algorithm’s input, 23, 849–850, 973–975
  of a binomial tree, 457
  of a boolean combinational circuit, 989
  of a clique, 1003
  of a comparison network, 707
  of a set, 1073
  of a sorting network, 708 ex.
  of a subtree in a Fibonacci heap, 495
  of a vertex cover, 1006, 1024
skew symmetry, 644
skip list, 301
slack, 781
slack form, 773, 781–783
  uniqueness of, 801
slack variable, 781
slot, 222
SLOW-ALL-PAIRS-SHORTEST-PATHS, 625
solution
  to an abstract problem, 972
  basic, 792
  to a computational problem, 6
  to a concrete problem, 973
  feasible, 601, 773, 778
  infeasible, 778
  optimal, 778
  to a system of linear equations, 742
sorted linked list, 204
see also linked list
SORTER, 719, 720 ex.
sorting, 15–19, 28–36, 123–182
  average-case lower bound for, 178 pr.
  bubblesort, 38 pr.
  bucket sort, 174–177
  comparison sort, 165
  counting sort, 168–170
  fuzzy, 163 pr.
  heapsort, 127–144
  insertion sort, 11, 15–19
  lexicographic, 269 pr.
  in linear time, 168–177, 178 pr.
  lower bounds for, 165–168
  of a matrix, 721 ex.
  merge sort, 11, 28–36
  network for, see sorting network
  in place, 16, 124
  of points by polar angle, 939 ex.
  problem of, 5, 15, 123
  quicksort, 145–164
  radix sort, 170–173
  selection sort, 27 ex.
  Shell’s sort, 40
  topological, see topological sort
  using a binary search tree, 264 ex.
  using networks, see sorting network
  variable-length items, 179 pr.
sorting network, 707
  AKS, 724
  based on insertion sort, 708 ex.
  based on merge sort, 719–721
  bitonic, 712–716
  depth of, 708 ex., 720 ex.
  odd-even, 721 pr.
  size of, 708 ex.
  source, 531, 581, 644, 647
  spanning tree, 394, 561
  bottleneck, 577 pr.
  verification of, 579
  see also minimum spanning tree
  sparse graph, 527
  sparse-hulled distribution, 964 pr.
  spindle, 435
  spine, 296 pr.
  splat tree, 301, 432
  spline, 767 pr.
  splitting
  of B-tree nodes, 443–445
  of 2-3-4 trees, 453 pr.
splitting summations, 1065–1066
  spurious hit, 912
  square matrix, 726
  square of a directed graph, 530 ex.
  square root, modulo a prime, 903 pr.
  squaring, repeated
    for all-pairs shortest paths, 625–627
    for raising a number to a power, 879
  stability
    numerical, 725, 743, 769
    of sorting algorithms, 170, 173 ex.
  stack, 200–201
    in Graham’s scan, 949
    implemented by queues, 204 ex.
    linked-list implementation of, 208 ex.
    operations analyzed by accounting method, 410–411
    operations analyzed by aggregate analysis, 406–408
    operations analyzed by potential method, 413–414
    for procedure execution, 162 pr.
    on secondary storage, 452 pr.
STACK-EMPTY, 201
  standard deviation, 1110
  standard form, 773, 777–781
  star-shaped polygon, 956 ex.
  start state, 916
  start time, 371
  state of a finite automaton, 916
  static graph, 499 n.
  static set of keys, 245
  Stirling’s approximation, 55
  STOOG-E-SORT, 161 pr.
  storage management, 127, 210–212, 213 ex., 229 ex.
  store instruction, 22
  straddle, 936
  Strassen’s algorithm, 735–742
  streaks, 110–114
  strictly decreasing, 51
  strictly increasing, 51
  string, 906, 1095
  string matching, 906–932
    based on repetition factors, 931 pr.
    by finite automata, 916–923
    with gap characters, 910 ex., 923 ex.
    Knuth-Morris-Pratt algorithm for, 923–931
    naive algorithm for, 909–911
Rabin-Karp algorithm for, 911–916
string-matching automaton, 917–922, 923 ex.
strongly connected component, 1082
decomposition into, 552–557
STRONGLY-CONNECTED-COMPONENTS, 554
strongly connected graph, 1082
subgraph, 1082
predecessor, see predecessor subgraph
subgraph-isomorphism problem, 1017 ex.
subgroup, 866–868
subpath, 1081
subroutine
calling, 20, 22, 23 n.
executing, 23 n.
subsequence, 350
subset
hereditary family of, 393
independent family of, 393
subset (⊆), 1071
SUBSET-SUM, 1013
subset-sum problem
approximation algorithm for, 1043–1049
NP-completeness of, 1013–1017
with unary target, 1017 ex.
substitution method, 63–67
and recursion trees, 70–72
substring, 1095
subtract instruction, 22
subtraction of matrices, 729
subtree, 1087
maintaining sizes of, in order-statistic trees, 306–307
success in a Bernoulli trial, 1112
successor
in binary search trees, 258–259
finding i th of a node in an order-statistic tree, 307 ex.
in linked lists, 204
in order-statistic trees, 310 ex.
in red-black trees, 276
SUCCESSOR, 198
suffix (□), 907
suffix function, 917
suffix-function inequality, 920
suffix-function recursion lemma, 920
sum
Cartesian, 830 ex.
flow, 650 ex.
infinite, 1058
of matrices, 728
of polynomials, 822
rule of, 1094
telescoping, 1061
summation, 1058–1069
in asymptotic notation, 47, 1059
bounding, 1062–1069
formulas and properties of, 1058–1062
implicit, 648
linearity of, 1059
summation lemma, 832
superpolynomial time, 966
supersink, 647
supersource, 647
surjection, 1078
SVD, 769
sweeping, 940–947, 962 pr.
sweep line, 940
sweep-line status, 942–943
symbol table, 221, 230, 232
symmetric difference, 696 pr.
symmetric matrix, 728, 733–734 ex.
symmetric positive-definite matrix, 760–762
symmetric relation, 1075
symmetry of Θ-notation, 49
systems of difference constraints, 601–607
systems of linear equations, 742–755, 767 pr.
TABLE-DELETE, 422
TABLE-INSERT, 418
tail
of a binomial distribution, 1118–1125
of a linked list, 204
of a queue, 202
tail recursion, 162 pr., 376
target, 1013
Tarjan’s off-line least-common-ancestors algorithm, 521 pr.
task, 399
task scheduling, 399–402, 404 pr.
tautology, 983 ex., 1002 ex.
Taylor series, 271 pr.
telescoping series, 1061
telescoping sum, 1061
testing
of primality, 887–896, 904
of pseudoprimality, 889–890
text in string matching, 906
then, in pseudocode, 19
3-CNF, 999
3-CNF-SAT, 999
3-CNF satisfiability, 998–1002
   approximation algorithm for, 1039–1040
   and 2-CNF satisfiability, 967
3-COLOR, 1019 pr.
3-conjunctive normal form, 999
   tight constraint, 791
time, see running time
time domain, 822
timestamp, 540, 548 ex.
Toeplitz matrix, 844 pr.
TOP, 949
top of a stack, 200
topological sort, 549–552
   in computing single-source shortest paths in a dag, 592
TOPOLOGICAL-SORT, 550
total net flow, 645
total order, 1077
total path length, 270 pr.
total positive flow, 645
tour
   bitonic, 364 pr.
   Euler, 559 pr., 966
   of a graph, 1012
track, 435
tractability, 966
transition function, 916, 921–922
transitive closure, 632–635
   and boolean matrix multiplication, 759 ex.
of dynamic graphs, 641 pr.
TRANSITIVE-CLOSURE, 633
transitive relation, 1075
transitivity of asymptotic notation, 49
transpose
   conjugate, 759 ex.
   of a directed graph, 530 ex.
   of a matrix, 529, 726
transpose symmetry of asymptotic notation, 49
transposition network, 721 pr.
travelling-salesman problem
   approximation algorithm for, 1027–1033
   bitonic euclidean, 364 pr.
bottleneck, 1033 ex.
NP-completeness of, 1012–1013
   with the triangle inequality, 1028–1031
   without the triangle inequality, 1031–1032
traversal of a tree, see tree walk
treap, 296 pr.
TREAP-INSERT, 296 pr.
tree, 1085–1091
   AA-trees, 301
   AVL, 296 pr.
binary, see binary tree
   binomial, 457–459, 479
   bisection of, 1092 pr.
breadth-first, 532, 538
   B-trees, 434–454
decision, 166–167
depth-first, 540
diameter of, 539 ex.
dynamic, 432
tree, 1083, 1085–1087
   full walk of, 1030
   fusion, 182, 433
   heap, 127–144
   height-balanced, 296 pr.
   height of, 1088
   interval, 311–317
   k-neighbor, 301
minimum spanning, see minimum spanning tree
   optimal binary search, 356–363, 369
order-statistic, 302–308
parse, 999
   recursion, 36, 67–72
red-black, see red-black tree
   rooted, 214–217, 1087
scapegoat, 301
search, see search tree
   shortest-paths, 584, 610–613
   spanning, see minimum spanning tree,
   spanning tree
   splay, 301, 432
treap, 296 pr.
   2-3, 300, 454
   2-3-4, 439, 453 pr.
   walk, see tree walk
weight-balanced trees, 301
TREE-DELETE, 262, 288
tree edge, 538, 540, 546
TREE-INSERT, 261, 280
TREE-MAXIMUM, 258
TREE-MINIMUM, 258
TREE-PREDECESSOR, 259
TREE-SEARCH, 257
TREE-SUCCESSOR, 259
tree walk, 254, 260 ex., 305, 1030
trial, Bernoulli, 1112
trial division, 888
triangle inequality, 1028
   and negative-weight edges, 1032 ex.
   for shortest paths, 587, 607–608
triangular matrix, 727–728, 733 ex.
trichotomy, interval, 311
trichotomy property of real numbers, 49
   and negative-weight edges, 1032 ex.
   for shortest paths, 587, 607–608
   – 608
triangular matrix, 727–728, 733 ex.
trie, see radix tree
TRIM, 1046
trimming of a list, 1045
trivial divisor, 851
truth assignment, 988, 996
truth table, 987
TSP, 1012
tuple, 1074
twiddle factor, 836
2-CNF-SAT, 1003 ex.
2-CNF satisfiability, 1003 ex.
   and 3-CNF satisfiability, 967
two-pass method, 508
2-3-4 heap, 473 pr.
2-3-4 tree, 439
   joining, 453 pr.
   and red-black trees, 441 ex.
   splitting, 453 pr.
2-3 tree, 300, 454
unary, 974
unbounded linear program, 778
unconditional branch instruction, 22
uncountable set, 1073
underdetermined system of linear equations, 743
underflow
   of a queue, 202
   of a stack, 201
undirected graph, 1080
   articulation point of, 558 pr.
   biconnected component of, 558 pr.
   bridge of, 558 pr.
   clique in, 1003
   coloring of, 1019 pr., 1091 pr.
   computing a minimum spanning tree in,
      561–579
   converting to, from a multigraph, 530 ex.
$d$-regular, 669 ex.
grid, 692 pr.
hamiltonian, 979
   independent set of, 1018 pr.
   matching of, 664
   nonhamiltonian, 979
   vertex cover of, 1006, 1024
see also graph
undirected version of a directed graph, 1082
uniform hashing, 239
uniform probability distribution, 1101–1102
uniform random permutation, 93, 101
union
   of dynamic sets, see uniting
   of languages, 976
   of sets ($\cup$), 1071
UNION, 455, 499
   disjoint-set-forest implementation of, 508
   linked-list implementation of, 502–504,
      505 ex.
union by rank, 506
unique factorization of integers, 854
unit (1), 851
uniting
   of binomial heaps, 462–468
   of Fibonacci heaps, 481–482
   of heaps, 455
   of linked lists, 208 ex.
   of 2-3-4 heaps, 473 pr.
unit lower-triangular matrix, 728
unit-time task, 399
unit upper-triangular matrix, 727
unit vector, 726
universal hashing, 232–236
universal sink, 530 ex.
universe, 1072
unmatched vertex, 664
unordered binomial tree, 479
unsorted linked list, 204
   see also linked list
unweighted longest simple paths, 342
unweighted shortest paths, 341
upper-bound property, 587, 608
upper median, 183
upper-triangular matrix, 727
valid shift, 906
value
   of a flow, 644
<table>
<thead>
<tr>
<th>Term</th>
<th>Page(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>of a function</td>
<td>1078</td>
</tr>
<tr>
<td>objective</td>
<td>774, 778</td>
</tr>
<tr>
<td>Vandermonde matrix</td>
<td>734 ex.</td>
</tr>
<tr>
<td>van Emde Boas data structure</td>
<td>144, 433</td>
</tr>
<tr>
<td>( \text{Var}[ \cdot ] ) (variance)</td>
<td>1110</td>
</tr>
<tr>
<td>variable</td>
<td></td>
</tr>
<tr>
<td>basic</td>
<td>782</td>
</tr>
<tr>
<td>entering</td>
<td>793</td>
</tr>
<tr>
<td>leaving</td>
<td>793</td>
</tr>
<tr>
<td>nonbasic</td>
<td>782</td>
</tr>
<tr>
<td>in pseudocode</td>
<td>19</td>
</tr>
<tr>
<td>random</td>
<td>1106–1111</td>
</tr>
<tr>
<td>slack</td>
<td>781</td>
</tr>
<tr>
<td>see also indicator random variable</td>
<td></td>
</tr>
<tr>
<td>variable-length code</td>
<td>385</td>
</tr>
<tr>
<td>variance</td>
<td>1109</td>
</tr>
<tr>
<td>of a binomial distribution</td>
<td>1115</td>
</tr>
<tr>
<td>of a geometric distribution</td>
<td>1112</td>
</tr>
<tr>
<td>vector</td>
<td>726, 730–731</td>
</tr>
<tr>
<td>convolution of</td>
<td>825</td>
</tr>
<tr>
<td>cross product of</td>
<td>934</td>
</tr>
<tr>
<td>orthonormal</td>
<td>769</td>
</tr>
<tr>
<td>in the plane</td>
<td>934</td>
</tr>
<tr>
<td>Venn diagram</td>
<td>1072</td>
</tr>
<tr>
<td>verification</td>
<td>979–983</td>
</tr>
<tr>
<td>of spanning trees</td>
<td>579</td>
</tr>
<tr>
<td>verification algorithm</td>
<td>980</td>
</tr>
<tr>
<td>vertex</td>
<td></td>
</tr>
<tr>
<td>articulation point</td>
<td>558 pr.</td>
</tr>
<tr>
<td>in a graph</td>
<td>1080</td>
</tr>
<tr>
<td>intermediate</td>
<td>629</td>
</tr>
<tr>
<td>isolated</td>
<td>1081</td>
</tr>
<tr>
<td>of a polygon</td>
<td>939 ex.</td>
</tr>
<tr>
<td>selector</td>
<td>1009</td>
</tr>
<tr>
<td>vertex cover</td>
<td>1006, 1024, 1040–1043</td>
</tr>
<tr>
<td>( \text{VERTEX-COVER} ), ( \text{VER} )</td>
<td>1006</td>
</tr>
<tr>
<td>vertex-cover problem</td>
<td></td>
</tr>
<tr>
<td>approximation algorithm for</td>
<td>1024–1027</td>
</tr>
<tr>
<td>NP-completeness of</td>
<td>1006–1008</td>
</tr>
<tr>
<td>vertex set</td>
<td>1080</td>
</tr>
<tr>
<td>violation of an equality constraint</td>
<td>791</td>
</tr>
<tr>
<td>virtual memory</td>
<td>22</td>
</tr>
<tr>
<td>Viterbi algorithm</td>
<td>367 pr.</td>
</tr>
<tr>
<td>VLSI (very large scale integration)</td>
<td>87 n.</td>
</tr>
<tr>
<td>walk of a tree</td>
<td></td>
</tr>
<tr>
<td>weak duality</td>
<td>805</td>
</tr>
<tr>
<td>weight</td>
<td></td>
</tr>
<tr>
<td>of a cut</td>
<td>1043 ex.</td>
</tr>
<tr>
<td>of an edge</td>
<td>529</td>
</tr>
<tr>
<td>mean</td>
<td>617 pr.</td>
</tr>
<tr>
<td>of a path</td>
<td>580</td>
</tr>
<tr>
<td>weight-balanced tree</td>
<td>301, 427 pr.</td>
</tr>
<tr>
<td>weighted bipartite matching</td>
<td>497</td>
</tr>
<tr>
<td>weighted matroid</td>
<td>394–398</td>
</tr>
<tr>
<td>weighted median</td>
<td>194 pr.</td>
</tr>
<tr>
<td>weighted set-covering problem</td>
<td>1050 pr.</td>
</tr>
<tr>
<td>weighted-union heuristic</td>
<td>503</td>
</tr>
<tr>
<td>weighted vertex cover</td>
<td>1040–1043</td>
</tr>
<tr>
<td>weight function</td>
<td></td>
</tr>
<tr>
<td>for a graph</td>
<td>529</td>
</tr>
<tr>
<td>in a weighted matroid</td>
<td>394</td>
</tr>
<tr>
<td>( \text{while} ), in pseudocode</td>
<td>19</td>
</tr>
<tr>
<td>white-path theorem</td>
<td>545</td>
</tr>
<tr>
<td>white vertex</td>
<td>531, 540</td>
</tr>
<tr>
<td>widget</td>
<td>1008</td>
</tr>
<tr>
<td>wire</td>
<td>705, 988</td>
</tr>
<tr>
<td>( \text{WITNESS} )</td>
<td>891</td>
</tr>
<tr>
<td>witness to the compositeness of a number</td>
<td>890</td>
</tr>
<tr>
<td>worst-case running time</td>
<td>26, 46</td>
</tr>
<tr>
<td>Yen’s improvement to the Bellman-Ford</td>
<td></td>
</tr>
<tr>
<td>algorithm</td>
<td>614 pr.</td>
</tr>
<tr>
<td>Young tableau</td>
<td>143 pr.</td>
</tr>
<tr>
<td>( \mathbb{Z} ) (set of integers)</td>
<td>1070</td>
</tr>
<tr>
<td>( \mathbb{Z}_n ) (equivalence classes modulo ( n ))</td>
<td>851</td>
</tr>
<tr>
<td>( \mathbb{Z}_n^\ast ) (elements of multiplicative group modulo ( n ))</td>
<td>864</td>
</tr>
<tr>
<td>( \mathbb{Z}_n^+ ) (nonzero elements of ( \mathbb{Z}_n^\ast ))</td>
<td>889</td>
</tr>
<tr>
<td>zero matrix</td>
<td>726</td>
</tr>
<tr>
<td>zero of a polynomial modulo a prime</td>
<td>872 ex.</td>
</tr>
<tr>
<td>0-1 integer-programming problem</td>
<td>1017 ex., 1041</td>
</tr>
<tr>
<td>0-1 knapsack problem</td>
<td>382, 384 ex.</td>
</tr>
<tr>
<td>zero-one principle</td>
<td>709–712, 716–717 ex.</td>
</tr>
</tbody>
</table>

VPL