

Index

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 or problem, which is not necessarily the page on which the reference actually appears.

- $\alpha(n)$, 574
- ϕ (golden ratio), 59, 108 pr.
- $\hat{\phi}$ (conjugate of the golden ratio), 59
- $\phi(n)$ (Euler’s phi function), 943
- $\rho(n)$ -approximation algorithm, 1106, 1123
- o -notation, 50–51, 64
- O -notation, 45 fig., 47–48, 64
- \mathcal{O}' -notation, 62 pr.
- \tilde{O} -notation, 62 pr.
- ω -notation, 51
- Ω -notation, 45 fig., 48–49, 64
- $\tilde{\Omega}$ -notation, 62 pr.
- $\tilde{\Omega}$ -notation, 62 pr.
- Θ -notation, 44–47, 45 fig., 64
- $\tilde{\Theta}$ -notation, 62 pr.
- $\{ \}$ (set), 1158
- \in (set member), 1158
- \notin (not a set member), 1158
- \emptyset
 - (empty language), 1058
 - (empty set), 1158
- \subseteq (subset), 1159
- \subset (proper subset), 1159
- $:$ (such that), 1159
- \cap (set intersection), 1159
- \cup (set union), 1159
- $-$ (set difference), 1159
- $| |$
 - (flow value), 710
 - (length of a string), 986
 - (set cardinality), 1161
- \times
 - (Cartesian product), 1162
 - (cross product), 1016
- $\langle \rangle$
 - (sequence), 1166
 - (standard encoding), 1057
- $\binom{n}{k}$ (choose), 1185
- $\| \|$ (euclidean norm), 1222
- $!$ (factorial), 57
- $\lceil \rceil$ (ceiling), 54
- $\lfloor \rfloor$ (floor), 54
- $\sqrt{}$ (lower square root), 546
- $\sqrt{}$ (upper square root), 546
- \sum (sum), 1145
- \prod (product), 1148
- \rightarrow (adjacency relation), 1169
- \rightsquigarrow (reachability relation), 1170
- \wedge (AND), 697, 1071
- \neg (NOT), 1071
- \vee (OR), 697, 1071
- \oplus (group operator), 939
- \otimes (convolution operator), 901

- * (closure operator), 1058
- | (divides relation), 927
- † (does-not-divide relation), 927
- \equiv (equivalent modulo n), 54, 1165 ex.
- $\not\equiv$ (not equivalent modulo n), 54
- $[a]_n$ (equivalence class modulo n), 928
- $+_n$ (addition modulo n), 940
- \cdot_n (multiplication modulo n), 940
- $(\frac{a}{p})$ (Legendre symbol), 982 pr.
- ε (empty string), 986, 1058
- \sqsubset (prefix relation), 986
- \sqsupset (suffix relation), 986
- \succsim_x (above relation), 1022
- // (comment symbol), 21
- \gg (much-greater-than relation), 574
- \ll (much-less-than relation), 783
- \leq_P (polynomial-time reducibility relation), 1067, 1077 ex.
- AA-tree, 338
- abelian group, 940
- ABOVE, 1024
- above relation (\succsim_x), 1022
- absent child, 1178
- absolutely convergent series, 1146
- absorption laws for sets, 1160
- abstract problem, 1054
- acceptable pair of integers, 972
- acceptance
 - by an algorithm, 1058
 - by a finite automaton, 996
- accepting state, 995
- accounting method, 456–459
 - for binary counters, 458
 - for dynamic tables, 465–466
 - for stack operations, 457–458, 458 ex.
- Ackermann's function, 585
- activity-selection problem, 415–422, 450
- acyclic graph, 1170
 - relation to matroids, 448 pr.
- add instruction, 23
- addition
 - of binary integers, 22 ex.
 - of matrices, 1220
 - modulo n ($+_n$), 940
 - of polynomials, 898
- additive group modulo n , 940
- addressing, open, *see* open-address hash table
- ADD-SUBARRAY, 805 pr.
- adjacency-list representation, 590
 - replaced by a hash table, 593 ex.
- adjacency-matrix representation, 591
- adjacency relation (\rightarrow), 1169
- adjacent vertices, 1169
- admissible edge, 749
- admissible network, 749–750
- adversary, 190
- aggregate analysis, 452–456
 - for binary counters, 454–455
 - for breadth-first search, 597
 - for depth-first search, 606
 - for Dijkstra's algorithm, 661
 - for disjoint-set data structures, 566–567, 568 ex.
 - for dynamic tables, 465
 - for Fibonacci heaps, 518, 522 ex.
 - for Graham's scan, 1036
 - for the Knuth-Morris-Pratt algorithm, 1006
 - for Prim's algorithm, 636
 - for rod-cutting, 367
 - for shortest paths in a dag, 655
 - for stack operations, 452–454
- aggregate flow, 863
- Akra-Bazzi method for solving a recurrence, 112–113
- algorithm, 5
 - correctness of, 6
 - origin of word, 42
 - running time of, 25
 - as a technology, 13
- Alice, 959
- ALLOCATE-NODE, 492
- ALLOCATE-OBJECT, 244
- allocation of objects, 243–244
- all-pairs shortest paths, 644, 684–707
 - in dynamic graphs, 707
 - in ϵ -dense graphs, 706 pr.
 - Floyd-Warshall algorithm for, 693–697, 706
 - Johnson's algorithm for, 700–706
 - by matrix multiplication, 686–693, 706–707
 - by repeated squaring, 689–691
- alphabet, 995, 1057
- $\alpha(n)$, 574
- amortized analysis, 451–478
 - accounting method of, 456–459
 - aggregate analysis, 367, 452–456

- for bit-reversal permutation, 472 pr.
- for breadth-first search, 597
- for depth-first search, 606
- for Dijkstra's algorithm, 661
- for disjoint-set data structures, 566–567, 568 ex., 572 ex., 575–581, 581–582 ex.
- for dynamic tables, 463–471
- for Fibonacci heaps, 509–512, 517–518, 520–522, 522 ex.
- for the generic push-relabel algorithm, 746
- for Graham's scan, 1036
- for the Knuth-Morris-Pratt algorithm, 1006
- for making binary search dynamic, 473 pr.
- potential method of, 459–463
- for restructuring red-black trees, 474 pr.
- for self-organizing lists with move-to-front, 476 pr.
- for shortest paths in a dag, 655
- for stacks on secondary storage, 502 pr.
- for weight-balanced trees, 473 pr.
- amortized cost
 - in the accounting method, 456
 - in aggregate analysis, 452
 - in the potential method, 459
- ancestor, 1176
 - least common, 584 pr.
- AND function (\wedge), 697, 1071
- AND gate, 1070
- and, in pseudocode, 22
- antiparallel edges, 711–712
- antisymmetric relation, 1164
- ANY-SEGMENTS-INTERSECT, 1025
- approximation
 - by least squares, 835–839
 - of summation by integrals, 1154–1156
- approximation algorithm, 10, 1105–1140
 - for bin packing, 1134 pr.
 - for MAX-CNF satisfiability, 1127 ex.
 - for maximum clique, 1111 ex., 1134 pr.
 - for maximum matching, 1135 pr.
 - for maximum spanning tree, 1137 pr.
 - for maximum-weight cut, 1127 ex.
 - for MAX-3-CNF satisfiability, 1123–1124, 1139
 - for minimum-weight vertex cover, 1124–1127, 1139
 - for parallel machine scheduling, 1136 pr.
 - randomized, 1123
 - for set cover, 1117–1122, 1139
 - for subset sum, 1128–1134, 1139
 - for traveling-salesman problem, 1111–1117, 1139
 - for vertex cover, 1108–1111, 1139
 - for weighted set cover, 1135 pr.
 - for 0-1 knapsack problem, 1137 pr., 1139
- approximation error, 836
- approximation ratio, 1106, 1123
- approximation scheme, 1107
- APPROX-MIN-WEIGHT-VC, 1126
- APPROX-SUBSET-SUM, 1131
- APPROX-TSP-TOUR, 1112
- APPROX-VERTEX-COVER, 1109
- arbitrage, 679 pr.
- arc, *see* edge
- argument of a function, 1166–1167
- arithmetic instructions, 23
- arithmetic, modular, 54, 939–946
- arithmetic series, 1146
- arithmetic with infinities, 650
- arm, 485
- array, 21
 - Monge, 110 pr.
 - passing as a parameter, 21
- articulation point, 621 pr.
- assignment
 - multiple, 21
 - satisfying, 1072, 1079
 - truth, 1072, 1079
- associative laws for sets, 1160
- associative operation, 939
- asymptotically larger, 52
- asymptotically nonnegative, 45
- asymptotically positive, 45
- asymptotically smaller, 52
- asymptotically tight bound, 45
- asymptotic efficiency, 43
- asymptotic lower bound, 48
- asymptotic notation, 43–53, 62 pr.
 - and graph algorithms, 588
 - and linearity of summations, 1146
- asymptotic upper bound, 47
- attribute of an object, 21
- augmentation of a flow, 716
- augmenting data structures, 339–355
- augmenting path, 719–720, 763 pr.
- authentication, 284 pr., 960–961, 964

- automaton
 - finite, 995
 - string-matching, 996–1002
- auxiliary hash function, 272
- auxiliary linear program, 886
- average-case running time, 28, 116
- AVL-INSERT, 333 pr.
- AVL tree, 333 pr., 337
- axioms, for probability, 1190
- babyface, 602 ex.
- back edge, 609, 613
- back substitution, 817
- BAD-SET-COVER-INSTANCE, 1122 ex.
- BALANCE, 333 pr.
- balanced search tree
 - AA-trees, 338
 - AVL trees, 333 pr., 337
 - B-trees, 484–504
 - k -neighbor trees, 338
 - red-black trees, 308–338
 - scapegoat trees, 338
 - splay trees, 338, 482
 - treaps, 333 pr., 338
 - 2-3-4 trees, 489, 503 pr.
 - 2-3 trees, 337, 504
 - weight-balanced trees, 338, 473 pr.
- balls and bins, 133–134, 1215 pr.
- base- a pseudoprime, 967
- base case, 65, 84
- base, in DNA, 391
- basic feasible solution, 866
- basic solution, 866
- basic variable, 855
- basis function, 835
- Bayes's theorem, 1194
- BELLMAN-FORD, 651
- Bellman-Ford algorithm, 651–655, 682
 - for all-pairs shortest paths, 684
 - in Johnson's algorithm, 702–704
 - and objective functions, 670 ex.
 - to solve systems of difference constraints, 668
 - Yen's improvement to, 678 pr.
- BELOW, 1024
- Bernoulli trial, 1201
 - and balls and bins, 133–134
 - and streaks, 135–139
- best-case running time, 29 ex., 49
- BFS, 595
- BIASED-RANDOM, 117 ex.
- biconnected component, 621 pr.
- big-oh notation, 45 fig., 47–48, 64
- big-omega notation, 45 fig., 48–49, 64
- bijective function, 1167
- binary character code, 428
- binary counter
 - analyzed by accounting method, 458
 - analyzed by aggregate analysis, 454–455
 - analyzed by potential method, 461–462
 - bit-reversed, 472 pr.
- binary entropy function, 1187
- binary gcd algorithm, 981 pr.
- binary heap, *see* heap
- binary relation, 1163
- binary search, 39 ex.
 - with fast insertion, 473 pr.
 - in insertion sort, 39 ex.
 - in multithreaded merging, 799–800
 - in searching B-trees, 499 ex.
- BINARY-SEARCH, 799
- binary search tree, 286–307
 - AA-trees, 338
 - AVL trees, 333 pr., 337
 - deletion from, 295–298, 299 ex.
 - with equal keys, 303 pr.
 - insertion into, 294–295
 - k -neighbor trees, 338
 - maximum key of, 291
 - minimum key of, 291
 - optimal, 397–404, 413
 - predecessor in, 291–292
 - querying, 289–294
 - randomly built, 299–303, 304 pr.
 - right-converting of, 314 ex.
 - scapegoat trees, 338
 - searching, 289–291
 - for sorting, 299 ex.
 - splay trees, 338
 - successor in, 291–292
 - and treaps, 333 pr.
 - weight-balanced trees, 338
 - see also* red-black tree
- binary-search-tree property, 287
 - in treaps, 333 pr.
 - vs. min-heap property, 289 ex.

- binary tree, 1177
 - full, 1178
 - number of different ones, 306 pr.
 - representation of, 246
 - superimposed upon a bit vector, 533–534
 - see also* binary search tree
- binomial coefficient, 1186–1187
- binomial distribution, 1203–1206
 - and balls and bins, 133
 - maximum value of, 1207 ex.
 - tails of, 1208–1215
- binomial expansion, 1186
- binomial heap, 527 pr.
- binomial tree, 527 pr.
- bin packing, 1134 pr.
- bipartite graph, 1172
 - corresponding flow network of, 732
 - d -regular, 736 ex.
 - and hypergraphs, 1173 ex.
- bipartite matching, 530, 732–736, 747 ex., 766
 - Hopcroft-Karp algorithm for, 763 pr.
- birthday paradox, 130–133, 142 ex.
- bisection of a tree, 1181 pr.
- bitonic euclidean traveling-salesman problem, 405 pr.
- bitonic sequence, 682 pr.
- bitonic tour, 405 pr.
- bit operation, 927
 - in Euclid's algorithm, 981 pr.
- bit-reversal permutation, 472 pr., 918
- BIT-REVERSE-COPY, 918
- bit-reversed binary counter, 472 pr.
- BIT-REVERSED-INCREMENT, 472 pr.
- bit vector, 255 ex., 532–536
- black-height, 309
- black vertex, 594, 603
- blocking flow, 765
- block structure in pseudocode, 20
- Bob, 959
- Boole's inequality, 1195 ex.
- boolean combinational circuit, 1071
- boolean combinational element, 1070
- boolean connective, 1079
- boolean formula, 1049, 1066 ex., 1079, 1086 ex.
- boolean function, 1187 ex.
- boolean matrix multiplication, 832 ex.
- Borůvka's algorithm, 641
- bottleneck spanning tree, 640 pr.
- bottleneck traveling-salesman problem, 1117 ex.
- bottom of a stack, 233
- BOTTOM-UP-CUT-ROD, 366
- bottom-up method, for dynamic programming, 365
- bound
 - asymptotically tight, 45
 - asymptotic lower, 48
 - asymptotic upper, 47
 - on binomial coefficients, 1186–1187
 - on binomial distributions, 1206
 - polylogarithmic, 57
 - on the tails of a binomial distribution, 1208–1215
 - see also* lower bounds
- boundary condition, in a recurrence, 67, 84
- boundary of a polygon, 1020 ex.
- bounding a summation, 1149–1156
- box, nesting, 678 pr.
- B^+ -tree, 488
- branching factor, in B-trees, 487
- branch instructions, 23
- breadth-first search, 594–602, 623
 - in maximum flow, 727–730, 766
 - and shortest paths, 597–600, 644
 - similarity to Dijkstra's algorithm, 662, 663 ex.
- breadth-first tree, 594, 600
- bridge, 621 pr.
- B^* -tree, 489 n.
- B-tree, 484–504
 - compared with red-black trees, 484, 490
 - creating, 492
 - deletion from, 499–502
 - full node in, 489
 - height of, 489–490
 - insertion into, 493–497
 - minimum degree of, 489
 - minimum key of, 497 ex.
 - properties of, 488–491
 - searching, 491–492
 - splitting a node in, 493–495
 - 2-3-4 trees, 489
- B-TREE-CREATE, 492
- B-TREE-DELETE, 499
- B-TREE-INSERT, 495

- B-TREE-INSERT-NONFULL, 496
- B-TREE-SEARCH, 492, 499 ex.
- B-TREE-SPLIT-CHILD, 494
- BUBBLESORT, 40 pr.
- bucket, 200
- bucket sort, 200–204
- BUCKET-SORT, 201
- BUILD-MAX-HEAP, 157
- BUILD-MAX-HEAP', 167 pr.
- BUILD-MIN-HEAP, 159
- butterfly operation, 915
- by**, in pseudocode, 21
- cache, 24, 449 pr.
- cache hit, 449 pr.
- cache miss, 449 pr.
- cache obliviousness, 504
- caching, off-line, 449 pr.
- call
 - in a multithreaded computation, 776
 - of a subroutine, 23, 25 n.
 - by value, 21
- call edge, 778
- cancellation lemma, 907
- cancellation of flow, 717
- canonical form for task scheduling, 444
- capacity
 - of a cut, 721
 - of an edge, 709
 - residual, 716, 719
 - of a vertex, 714 ex.
- capacity constraint, 709–710
- cardinality of a set ($| \cdot |$), 1161
- Carmichael number, 968, 975 ex.
- Cartesian product (\times), 1162
- Cartesian sum, 906 ex.
- cascading cut, 520
- CASCADING-CUT, 519
- Catalan numbers, 306 pr., 372
- ceiling function ($\lceil \cdot \rceil$), 54
 - in master theorem, 103–106
- ceiling instruction, 23
- certain event, 1190
- certificate
 - in a cryptosystem, 964
 - for verification algorithms, 1063
- CHAINED-HASH-DELETE, 258
- CHAINED-HASH-INSERT, 258
- CHAINED-HASH-SEARCH, 258
- chaining, 257–260, 283 pr.
- chain of a convex hull, 1038
- changing a key, in a Fibonacci heap, 529 pr.
- changing variables, in the substitution method, 86–87
- character code, 428
- chess-playing program, 790–791
- child
 - in a binary tree, 1178
 - in a multithreaded computation, 776
 - in a rooted tree, 1176
- child list in a Fibonacci heap, 507
- Chinese remainder theorem, 950–954, 983
- chip multiprocessor, 772
- chirp transform, 914 ex.
- choose $\binom{n}{k}$, 1185
- chord, 345 ex.
- Cilk, 774, 812
- Cilk++, 774, 812
- ciphertext, 960
- circuit
 - boolean combinational, 1071
 - depth of, 919
 - for fast Fourier transform, 919–920
- CIRCUIT-SAT, 1072
- circuit satisfiability, 1070–1077
- circular, doubly linked list with a sentinel, 239
- circular linked list, 236
 - see also* linked list
- class
 - complexity, 1059
 - equivalence, 1164
- classification of edges
 - in breadth-first search, 621 pr.
 - in depth-first search, 609–610, 611 ex.
 - in a multithreaded dag, 778–779
- clause, 1081–1082
- clean area, 208 pr.
- clique, 1086–1089, 1105
 - approximation algorithm for, 1111 ex., 1134 pr.
- CLIQUE, 1087
- closed interval, 348
- closed semiring, 707
- closest pair, finding, 1039–1044, 1047
- closest-point heuristic, 1117 ex.

- closure
 - group property, 939
 - of a language, 1058
 - operator (*), 1058
 - transitive, *see* transitive closure
- cluster
 - in a bit vector with a superimposed tree of constant height, 534
 - for parallel computing, 772
 - in proto van Emde Boas structures, 538
 - in van Emde Boas trees, 546
- clustering, 272
- CNF (conjunctive normal form), 1049, 1082
- CNF satisfiability, 1127 ex.
- coarsening leaves of recursion
 - in merge sort, 39 pr.
 - when recursively spawning, 787
- code, 428–429
 - Huffman, 428–437, 450
- codeword, 429
- codomain, 1166
- coefficient
 - binomial, 1186
 - of a polynomial, 55, 898
 - in slack form, 856
- coefficient representation, 900
 - and fast multiplication, 903–905
- cofactor, 1224
- coin changing, 446 pr.
- colinearity, 1016
- collision, 257
 - resolution by chaining, 257–260
 - resolution by open addressing, 269–277
- collision-resistant hash function, 964
- coloring, 1103 pr., 1180 pr.
- color, of a red-black-tree node, 308
- column-major order, 208 pr.
- column rank, 1223
- columnsort, 208 pr.
- column vector, 1218
- combination, 1185
- combinational circuit, 1071
- combinational element, 1070
- combine step, in divide-and-conquer, 30, 65
- comment, in pseudocode (//), 21
- commodity, 862
- common divisor, 929
 - greatest, *see* greatest common divisor
- common multiple, 939 ex.
- common subexpression, 915
- common subsequence, 7, 391
 - longest, 7, 390–397, 413
- commutative laws for sets, 1159
- commutative operation, 940
- COMPACTIFY-LIST, 245 ex.
- compact list, 250 pr.
- COMPACT-LIST-SEARCH, 250 pr.
- COMPACT-LIST-SEARCH', 251 pr.
- comparable line segments, 1022
- COMPARE-EXCHANGE, 208 pr.
- compare-exchange operation, 208 pr.
- comparison sort, 191
 - and binary search trees, 289 ex.
 - randomized, 205 pr.
 - and selection, 222
- compatible activities, 415
- compatible matrices, 371, 1221
- competitive analysis, 476 pr.
- complement
 - of an event, 1190
 - of a graph, 1090
 - of a language, 1058
 - Schur, 820, 834
 - of a set, 1160
- complementary slackness, 894 pr.
- complete graph, 1172
- complete k -ary tree, 1179
 - see also* heap
- completeness of a language, 1077 ex.
- complete step, 782
- completion time, 447 pr., 1136 pr.
- complexity class, 1059
 - co-NP, 1064
 - NP, 1049, 1064
 - NPC, 1050, 1069
 - P, 1049, 1055
- complexity measure, 1059
- complex numbers
 - inverting matrices of, 832 ex.
 - multiplication of, 83 ex.
- complex root of unity, 906
 - interpolation at, 912–913
- component
 - biconnected, 621 pr.
 - connected, 1170
 - strongly connected, 1171

- component graph, 617
- composite number, 928
 - witness to, 968
- composition, of multithreaded computations, 784 fig.
- computational depth, 812
- computational geometry, 1014–1047
- computational problem, 5–6
- computation dag, 777
- computation, multithreaded, 777
- COMPUTE-PREFIX-FUNCTION, 1006
- COMPUTE-TRANSITION-FUNCTION, 1001
- concatenation
 - of languages, 1058
 - of strings, 986
- concrete problem, 1055
- concurrency keywords, 774, 776, 785
- concurrency platform, 773
- conditional branch instruction, 23
- conditional independence, 1195 ex.
- conditional probability, 1192, 1194
- configuration, 1074
- conjugate of the golden ratio ($\hat{\phi}$), 59
- conjugate transpose, 832 ex.
- conjunctive normal form, 1049, 1082
- connected component, 1170–1171
 - identified using depth-first search, 612 ex.
 - identified using disjoint-set data structures, 562–564
- CONNECTED-COMPONENTS, 563
- connected graph, 1170
- connective, 1079
- co-NP (complexity class), 1064
- conquer step, in divide-and-conquer, 30, 65
- conservation of flow, 709–710
- consistency
 - of literals, 1088
 - sequential, 779, 812
- CONSOLIDATE, 516
- consolidating a Fibonacci-heap root list, 513–517
- constraint, 851
 - difference, 665
 - equality, 670 ex., 852–853
 - inequality, 852–853
 - linear, 846
 - nonnegativity, 851, 853
 - tight, 865
 - violation of, 865
- constraint graph, 666–668
- contain, in a path, 1170
- continuation edge, 778
- continuous uniform probability distribution, 1192
- contraction
 - of a dynamic table, 467–471
 - of a matroid, 442
 - of an undirected graph by an edge, 1172
- control instructions, 23
- convergence property, 650, 672–673
- convergent series, 1146
- converting binary to decimal, 933 ex.
- convex combination of points, 1015
- convex function, 1199
- convex hull, 8, 1029–1039, 1046 pr.
- convex layers, 1044 pr.
- convex polygon, 1020 ex.
- convex set, 714 ex.
- convolution (\otimes), 901
- convolution theorem, 913
- copy instruction, 23
- correctness of an algorithm, 6
- corresponding flow network for bipartite matching, 732
- countably infinite set, 1161
- counter, *see* binary counter
- counting, 1183–1189
 - probabilistic, 143 pr.
- counting sort, 194–197
 - in radix sort, 198
- COUNTING-SORT, 195
- coupon collector's problem, 134
- cover
 - path, 761 pr.
 - by a subset, 1118
 - vertex, 1089, 1108, 1124–1127, 1139
- covertical, 1024
- CREATE-NEW-RS-VEB-TREE, 557 pr.
- credit, 456
- critical edge, 729
- critical path
 - of a dag, 657
 - of a multithreaded computation, 779
- cross a cut, 626
- cross edge, 609
- cross product (\times), 1016

- cryptosystem, 958–965, 983
- cubic spline, 840 pr.
- currency exchange, 390 ex., 679 pr.
- curve fitting, 835–839
- cut
 - capacity of, 721
 - cascading, 520
 - of a flow network, 720–724
 - minimum, 721, 731 ex.
 - net flow across, 720
 - of an undirected graph, 626
 - weight of, 1127 ex.
- CUT, 519
- CUT-ROD, 363
- cutting, in a Fibonacci heap, 519
- cycle of a graph, 1170
 - hamiltonian, 1049, 1061
 - minimum mean-weight, 680 pr.
 - negative-weight, *see* negative-weight cycle and shortest paths, 646–647
- cyclic group, 955
- cyclic rotation, 1012 ex.
- cycling, of simplex algorithm, 875
- dag, *see* directed acyclic graph
- DAG-SHORTEST-PATHS, 655
- d -ary heap, 167 pr.
 - in shortest-paths algorithms, 706 pr.
- data-movement instructions, 23
- data-parallel model, 811
- data structure, 9, 229–355, 481–585
 - AA-trees, 338
 - augmentation of, 339–355
 - AVL trees, 333 pr., 337
 - binary search trees, 286–307
 - binomial heaps, 527 pr.
 - bit vectors, 255 ex., 532–536
 - B-trees, 484–504
 - deques, 236 ex.
 - dictionaries, 229
 - direct-address tables, 254–255
 - for disjoint sets, 561–585
 - for dynamic graphs, 483
 - dynamic sets, 229–231
 - dynamic trees, 482
 - exponential search trees, 212, 483
 - Fibonacci heaps, 505–530
 - fusion trees, 212, 483
 - hash tables, 256–261
 - heaps, 151–169
 - interval trees, 348–354
 - k -neighbor trees, 338
 - linked lists, 236–241
 - mergeable heap, 505
 - order-statistic trees, 339–345
 - persistent, 331 pr., 482
 - potential of, 459
 - priority queues, 162–166
 - proto van Emde Boas structures, 538–545
 - queues, 232, 234–235
 - radix trees, 304 pr.
 - red-black trees, 308–338
 - relaxed heaps, 530
 - rooted trees, 246–249
 - scapegoat trees, 338
 - on secondary storage, 484–487
 - skip lists, 338
 - splay trees, 338, 482
 - stacks, 232–233
 - treaps, 333 pr., 338
 - 2-3-4 heaps, 529 pr.
 - 2-3-4 trees, 489, 503 pr.
 - 2-3 trees, 337, 504
 - van Emde Boas trees, 531–560
 - weight-balanced trees, 338
- data type, 23
- deadline, 444
- deallocation of objects, 243–244
- decision by an algorithm, 1058–1059
- decision problem, 1051, 1054
 - and optimization problems, 1051
- decision tree, 192–193
- DECREASE-KEY, 162, 505
- decreasing a key
 - in Fibonacci heaps, 519–522
 - in 2-3-4 heaps, 529 pr.
- DECREMENT, 456 ex.
- degeneracy, 874
- degree
 - of a binomial-tree root, 527 pr.
 - maximum, of a Fibonacci heap, 509, 523–526
 - minimum, of a B-tree, 489
 - of a node, 1177
 - of a polynomial, 55, 898
 - of a vertex, 1169

- degree-bound, 898
- DELETE, 230, 505
- DELETE-LARGER-HALF, 463 ex.
- deletion
 - from binary search trees, 295–298, 299 ex.
 - from a bit vector with a superimposed binary tree, 534
 - from a bit vector with a superimposed tree of constant height, 535
 - from B-trees, 499–502
 - from chained hash tables, 258
 - from direct-address tables, 254
 - from dynamic tables, 467–471
 - from Fibonacci heaps, 522, 526 pr.
 - from heaps, 166 ex.
 - from interval trees, 349
 - from linked lists, 238
 - from open-address hash tables, 271
 - from order-statistic trees, 343–344
 - from proto van Emde Boas structures, 544
 - from queues, 234
 - from red-black trees, 323–330
 - from stacks, 232
 - from sweep-line statuses, 1024
 - from 2-3-4 heaps, 529 pr.
 - from van Emde Boas trees, 554–556
- DeMorgan's laws
 - for propositional logic, 1083
 - for sets, 1160, 1162 ex.
- dense graph, 589
- ϵ -dense, 706 pr.
- density
 - of prime numbers, 965–966
 - of a rod, 370 ex.
- dependence
 - and indicator random variables, 119
 - linear, 1223
 - see also* independence
- depth
 - average, of a node in a randomly built binary search tree, 304 pr.
 - of a circuit, 919
 - of a node in a rooted tree, 1177
 - of quicksort recursion tree, 178 ex.
 - of a stack, 188 pr.
- depth-determination problem, 583 pr.
- depth-first forest, 603
- depth-first search, 603–612, 623
 - in finding articulation points, bridges, and biconnected components, 621 pr.
 - in finding strongly connected components, 615–621, 623
 - in topological sorting, 612–615
- depth-first tree, 603
- deque, 236 ex.
- DEQUEUE, 235
- derivative of a series, 1147
- descendant, 1176
- destination vertex, 644
- det, *see* determinant
- determinacy race, 788
- determinant, 1224–1225
 - and matrix multiplication, 832 ex.
- deterministic algorithm, 123
 - multithreaded, 787
- DETERMINISTIC-SEARCH, 143 pr.
- DFS, 604
- DFS-VISIT, 604
- DFT (discrete Fourier transform), 9, 909
- diagonal matrix, 1218
 - LUP decomposition of, 827 ex.
- diameter of a tree, 602 ex.
- dictionary, 229
- difference constraints, 664–670
- difference equation, *see* recurrence
- difference of sets ($-$), 1159
 - symmetric, 763 pr.
- differentiation of a series, 1147
- digital signature, 960
- digraph, *see* directed graph
- DIJKSTRA, 658
- Dijkstra's algorithm, 658–664, 682
 - for all-pairs shortest paths, 684, 704
 - implemented with a Fibonacci heap, 662
 - implemented with a min-heap, 662
 - with integer edge weights, 664 ex.
 - in Johnson's algorithm, 702
 - similarity to breadth-first search, 662, 663 ex.
 - similarity to Prim's algorithm, 634, 662
- DIRECT-ADDRESS-DELETE, 254
- direct addressing, 254–255, 532–536
- DIRECT-ADDRESS-INSERT, 254
- DIRECT-ADDRESS-SEARCH, 254
- direct-address table, 254–255
- directed acyclic graph (dag), 1172

- and back edges, 613
- and component graphs, 617
- and hamiltonian paths, 1066 ex.
- longest simple path in, 404 pr.
- for representing a multithreaded
 - computation, 777
- single-source shortest-paths algorithm for, 655–658
- topological sort of, 612–615, 623
- directed graph, 1168
 - all-pairs shortest paths in, 684–707
 - constraint graph, 666
 - Euler tour of, 623 pr., 1048
 - hamiltonian cycle of, 1049
 - and longest paths, 1048
 - path cover of, 761 pr.
 - PERT chart, 657, 657 ex.
 - semiconnected, 621 ex.
 - shortest path in, 643
 - single-source shortest paths in, 643–683
 - singly connected, 612 ex.
 - square of, 593 ex.
 - transitive closure of, 697
 - transpose of, 592 ex.
 - universal sink in, 593 ex.
 - see also* directed acyclic graph, graph, network
- directed segment, 1015–1017
- directed version of an undirected graph, 1172
- DIRECTION, 1018
- dirty area, 208 pr.
- DISCHARGE, 751
- discharge of an overflowing vertex, 751
- discovered vertex, 594, 603
- discovery time, in depth-first search, 605
- discrete Fourier transform, 9, 909
- discrete logarithm, 955
- discrete logarithm theorem, 955
- discrete probability distribution, 1191
- discrete random variable, 1196–1201
- disjoint-set data structure, 561–585
 - analysis of, 575–581, 581 ex.
 - in connected components, 562–564
 - in depth determination, 583 pr.
 - disjoint-set-forest implementation of, 568–572
 - in Kruskal's algorithm, 631
 - linear-time special case of, 585
 - linked-list implementation of, 564–568
 - in off-line least common ancestors, 584 pr.
 - in off-line minimum, 582 pr.
 - in task scheduling, 448 pr.
- disjoint-set forest, 568–572
 - analysis of, 575–581, 581 ex.
 - rank properties of, 575, 581 ex.
 - see also* disjoint-set data structure
- disjoint sets, 1161
- disjunctive normal form, 1083
- disk, 1028 ex.
- disk drive, 485–487
 - see also* secondary storage
- DISK-READ, 487
- DISK-WRITE, 487
- distance
 - edit, 406 pr.
 - euclidean, 1039
 - L_m , 1044 ex.
 - Manhattan, 225 pr., 1044 ex.
 - of a shortest path, 597
- distributed memory, 772
- distribution
 - binomial, 1203–1206
 - continuous uniform, 1192
 - discrete, 1191
 - geometric, 1202–1203
 - of inputs, 116, 122
 - of prime numbers, 965
 - probability, 1190
 - sparse-hulled, 1046 pr.
 - uniform, 1191
- distributive laws for sets, 1160
- divergent series, 1146
- divide-and-conquer method, 30–35, 65
 - analysis of, 34–35
 - for binary search, 39 ex.
 - for conversion of binary to decimal, 933 ex.
 - for fast Fourier transform, 909–912
 - for finding the closest pair of points, 1040–1043
 - for finding the convex hull, 1030
 - for matrix inversion, 829–831
 - for matrix multiplication, 76–83, 792–797
 - for maximum-subarray problem, 68–75
 - for merge sort, 30–37, 797–805
 - for multiplication, 920 pr.

- for multithreaded matrix multiplication, 792–797
 - for multithreaded merge sort, 797–805
 - for quicksort, 170–190
 - relation to dynamic programming, 359
 - for selection, 215–224
 - solving recurrences for, 83–106, 112–113
 - for Strassen’s algorithm, 79–83
- divide instruction, 23
- divides relation (\mid), 927
- divide step, in divide-and-conquer, 30, 65
- division method, 263, 268–269 ex.
- division theorem, 928
- divisor, 927–928
 - common, 929
 - see also* greatest common divisor
- DNA, 6–7, 390–391, 406 pr.
- DNF (disjunctive normal form), 1083
- does-not-divide relation (\nmid), 927
- domain, 1166
- dominates relation, 1045 pr.
- double hashing, 272–274, 277 ex.
- doubly linked list, 236
 - see also* linked list
- downto**, in pseudocode, 21
- d -regular graph, 736 ex.
- duality, 879–886, 895 pr.
 - weak, 880–881, 886 ex.
- dual linear program, 879
- dummy key, 397
- dynamic graph, 562 n.
 - all-pairs shortest paths algorithms for, 707
 - data structures for, 483
 - minimum-spanning-tree algorithm for, 637 ex.
 - transitive closure of, 705 pr., 707
- dynamic multithreaded algorithm, *see* multithreaded algorithm
- dynamic multithreading, 773
- dynamic order statistics, 339–345
- dynamic-programming method, 359–413
 - for activity selection, 421 ex.
 - for all-pairs shortest paths, 686–697
 - for bitonic euclidean traveling-salesman problem, 405 pr.
 - bottom-up, 365
 - for breaking a string, 410 pr.
 - compared with greedy algorithms, 381, 390 ex., 418, 423–427
 - for edit distance, 406 pr.
 - elements of, 378–390
 - for Floyd-Warshall algorithm, 693–697
 - for inventory planning, 411 pr.
 - for longest common subsequence, 390–397
 - for longest palindrome subsequence, 405 pr.
 - for longest simple path in a weighted directed acyclic graph, 404 pr.
 - for matrix-chain multiplication, 370–378
 - and memoization, 387–389
 - for optimal binary search trees, 397–404
 - optimal substructure in, 379–384
 - overlapping subproblems in, 384–386
 - for printing neatly, 405 pr.
 - reconstructing an optimal solution in, 387
 - relation to divide-and-conquer, 359
 - for rod-cutting, 360–370
 - for seam carving, 409 pr.
 - for signing free agents, 411 pr.
 - top-down with memoization, 365
 - for transitive closure, 697–699
 - for Viterbi algorithm, 408 pr.
 - for 0-1 knapsack problem, 427 ex.
- dynamic set, 229–231
 - see also* data structure
- dynamic table, 463–471
 - analyzed by accounting method, 465–466
 - analyzed by aggregate analysis, 465
 - analyzed by potential method, 466–471
 - load factor of, 463
- dynamic tree, 482
- e , 55
- $E[]$ (expected value), 1197
- early-first form, 444
- early task, 444
- edge, 1168
 - admissible, 749
 - antiparallel, 711–712
 - attributes of, 592
 - back, 609
 - bridge, 621 pr.
 - call, 778
 - capacity of, 709
 - classification in breadth-first search, 621 pr.
 - classification in depth-first search, 609–610

- continuation, 778
- critical, 729
- cross, 609
- forward, 609
- inadmissible, 749
- light, 626
- negative-weight, 645–646
- residual, 716
- return, 779
- safe, 626
- saturated, 739
- spawn, 778
- tree, 601, 603, 609
- weight of, 591
- edge connectivity, 731 ex.
- edge set, 1168
- edit distance, 406 pr.
- Edmonds-Karp algorithm, 727–730
- elementary event, 1189
- elementary insertion, 465
- element of a set (\in), 1158
- ellipsoid algorithm, 850, 897
- elliptic-curve factorization method, 984
- elseif**, in pseudocode, 20 n.
- else**, in pseudocode, 20
- empty language (\emptyset), 1058
- empty set (\emptyset), 1158
- empty set laws, 1159
- empty stack, 233
- empty string (ϵ), 986, 1058
- empty tree, 1178
- encoding of problem instances, 1055–1057
- endpoint
 - of an interval, 348
 - of a line segment, 1015
- ENQUEUE, 235
- entering a vertex, 1169
- entering variable, 867
- entropy function, 1187
- ϵ -dense graph, 706 pr.
- ϵ -universal hash function, 269 ex.
- equality
 - of functions, 1166
 - linear, 845
 - of sets, 1158
- equality constraint, 670 ex., 852
 - and inequality constraints, 853
 - tight, 865
 - violation of, 865
- equation
 - and asymptotic notation, 49–50
 - normal, 837
 - recurrence, *see* recurrence
- equivalence class, 1164
 - modulo n ($[a]_n$), 928
- equivalence, modular (\equiv), 54, 1165 ex.
- equivalence relation, 1164
 - and modular equivalence, 1165 ex.
- equivalent linear programs, 852
- error**, in pseudocode, 22
- escape problem, 760 pr.
- EUCLID, 935
- Euclid's algorithm, 933–939, 981 pr., 983
- euclidean distance, 1039
- euclidean norm ($\| \cdot \|$), 1222
- Euler's constant, 943
- Euler's phi function, 943
- Euler's theorem, 954, 975 ex.
- Euler tour, 623 pr., 1048
 - and hamiltonian cycles, 1048
- evaluation of a polynomial, 41 pr., 900, 905 ex.
 - derivatives of, 922 pr.
 - at multiple points, 923 pr.
- event, 1190
- event point, 1023
- event-point schedule, 1023
- EXACT-SUBSET-SUM, 1129
- excess flow, 736
- exchange property, 437
- exclusion and inclusion, 1163 ex.
- execute a subroutine, 25 n.
- expansion of a dynamic table, 464–467
- expectation, *see* expected value
- expected running time, 28, 117
- expected value, 1197–1199
 - of a binomial distribution, 1204
 - of a geometric distribution, 1202
 - of an indicator random variable, 118
- explored vertex, 605
- exponential function, 55–56
- exponential height, 300
- exponential search tree, 212, 483
- exponential series, 1147
- exponentiation instruction, 24
- exponentiation, modular, 956
- EXTENDED-BOTTOM-UP-CUT-ROD, 369

- EXTENDED-EUCLID, 937
- EXTEND-SHORTEST-PATHS, 688
- extension of a set, 438
- exterior of a polygon, 1020 ex.
- external node, 1176
- external path length, 1180 ex.
- extracting the maximum key
 - from d -ary heaps, 167 pr.
 - from max-heaps, 163
- extracting the minimum key
 - from Fibonacci heaps, 512–518
 - from 2-3-4 heaps, 529 pr.
 - from Young tableaux, 167 pr.
- EXTRACT-MAX, 162–163
- EXTRACT-MIN, 162, 505
- factor, 928
 - twiddle, 912
- factorial function (!), 57–58
- factorization, 975–980, 984
 - unique, 931
- failure, in a Bernoulli trial, 1201
- fair coin, 1191
- fan-out, 1071
- Farkas's lemma, 895 pr.
- farthest-pair problem, 1030
- FASTER-ALL-PAIRS-SHORTEST-PATHS, 691, 692 ex.
- fast Fourier transform (FFT), 898–925
 - circuit for, 919–920
 - iterative implementation of, 915–918
 - multidimensional, 921 pr.
 - multithreaded algorithm for, 804 ex.
 - recursive implementation of, 909–912
 - using modular arithmetic, 923 pr.
- feasibility problem, 665, 894 pr.
- feasible linear program, 851
- feasible region, 847
- feasible solution, 665, 846, 851
- Fermat's theorem, 954
- FFT, *see* fast Fourier transform
- FFTW, 924
- FIB, 775
- FIB-HEAP-CHANGE-KEY, 529 pr.
- FIB-HEAP-DECREASE-KEY, 519
- FIB-HEAP-DELETE, 522
- FIB-HEAP-EXTRACT-MIN, 513
- FIB-HEAP-INSERT, 510
- FIB-HEAP-LINK, 516
- FIB-HEAP-PRUNE, 529 pr.
- FIB-HEAP-UNION, 512
- Fibonacci heap, 505–530
 - changing a key in, 529 pr.
 - compared with binary heaps, 506–507
 - creating, 510
 - decreasing a key in, 519–522
 - deletion from, 522, 526 pr.
 - in Dijkstra's algorithm, 662
 - extracting the minimum key from, 512–518
 - insertion into, 510–511
 - in Johnson's algorithm, 704
 - maximum degree of, 509, 523–526
 - minimum key of, 511
 - potential function for, 509
 - in Prim's algorithm, 636
 - pruning, 529 pr.
 - running times of operations on, 506 fig.
 - uniting, 511–512
- Fibonacci numbers, 59–60, 108 pr., 523
 - computation of, 774–780, 981 pr.
- FIFO (first-in, first-out), 232
 - see also* queue
- final-state function, 996
- final strand, 779
- FIND-DEPTH, 583 pr.
- FIND-MAX-CROSSING-SUBARRAY, 71
- FIND-MAXIMUM-SUBARRAY, 72
- find path, 569
- FIND-SET, 562
 - disjoint-set-forest implementation of, 571, 585
 - linked-list implementation of, 564
- finished vertex, 603
- finishing time, in depth-first search, 605
 - and strongly connected components, 618
- finish time, in activity selection, 415
- finite automaton, 995
 - for string matching, 996–1002
- FINITE-AUTOMATON-MATCHER, 999
- finite group, 940
- finite sequence, 1166
- finite set, 1161
- first-fit heuristic, 1134 pr.
- first-in, first-out, 232
 - see also* queue
- fixed-length code, 429

- floating-point data type, 23
- floor function ($\lfloor \cdot \rfloor$), 54
 - in master theorem, 103–106
- floor instruction, 23
- flow, 709–714
 - aggregate, 863
 - augmentation of, 716
 - blocking, 765
 - cancellation of, 717
 - excess, 736
 - integer-valued, 733
 - net, across a cut, 720
 - value of, 710
- flow conservation, 709–710
- flow network, 709–714
 - corresponding to a bipartite graph, 732
 - cut of, 720–724
 - with multiple sources and sinks, 712
- FLOYD-WARSHALL, 695
- FLOYD-WARSHALL', 699 ex.
- Floyd-Warshall algorithm, 693–697, 699–700 ex., 706
 - multithreaded, 797 ex.
- FORD-FULKERSON, 724
- Ford-Fulkerson method, 714–731, 765
- FORD-FULKERSON-METHOD, 715
- forest, 1172–1173
 - depth-first, 603
 - disjoint-set, 568–572
- for**, in pseudocode, 20–21
 - and loop invariants, 19 n.
- formal power series, 108 pr.
- formula satisfiability, 1079–1081, 1105
- forward edge, 609
- forward substitution, 816–817
- Fourier transform, *see* discrete Fourier transform, fast Fourier transform
- fractional knapsack problem, 426, 428 ex.
- free agent, 411 pr.
- freeing of objects, 243–244
- free list, 243
- FREE-OBJECT, 244
- free tree, 1172–1176
- frequency domain, 898
- full binary tree, 1178, 1180 ex.
 - relation to optimal code, 430
- full node, 489
- full rank, 1223
- full walk of a tree, 1114
- fully parenthesized matrix-chain product, 370
- fully polynomial-time approximation scheme, 1107
 - for subset sum, 1128–1134, 1139
- function, 1166–1168
 - Ackermann's, 585
 - basis, 835
 - convex, 1199
 - final-state, 996
 - hash, *see* hash function
 - linear, 26, 845
 - objective, 664, 847, 851
 - potential, 459
 - prefix, 1003–1004
 - quadratic, 27
 - reduction, 1067
 - suffix, 996
 - transition, 995, 1001–1002, 1012 ex.
- functional iteration, 58
- fundamental theorem of linear programming, 892
- furthest-in-future strategy, 449 pr.
- fusion tree, 212, 483
- fuzzy sorting, 189 pr.
- Gabow's scaling algorithm for single-source shortest paths, 679 pr.
- gap character, 989 ex., 1002 ex.
- gap heuristic, 760 ex., 766
- garbage collection, 151, 243
- gate, 1070
- Gaussian elimination, 819, 842
- gcd, *see* greatest common divisor
- general number-field sieve, 984
- generating function, 108 pr.
- generator
 - of a subgroup, 944
 - of \mathbb{Z}_n^* , 955
- GENERIC-MST, 626
- GENERIC-PUSH-RELABEL, 741
- generic push-relabel algorithm, 740–748
- geometric distribution, 1202–1203
 - and balls and bins, 134
- geometric series, 1147
- geometry, computational, 1014–1047
- $GF(2)$, 1227 pr.
- gift wrapping, 1037, 1047

- global variable, 21
- Goldberg's algorithm, *see* push-relabel algorithm
- golden ratio (ϕ), 59, 108 pr.
- gossiping, 478
- GRAFT, 583 pr.
- Graham's scan, 1030–1036, 1047
- GRAHAM-SCAN, 1031
- graph, 1168–1173
 - adjacency-list representation of, 590
 - adjacency-matrix representation of, 591
 - algorithms for, 587–766
 - and asymptotic notation, 588
 - attributes of, 588, 592
 - breadth-first search of, 594–602, 623
 - coloring of, 1103 pr.
 - complement of, 1090
 - component, 617
 - constraint, 666–668
 - dense, 589
 - depth-first search of, 603–612, 623
 - dynamic, 562 n.
 - ϵ -dense, 706 pr.
 - hamiltonian, 1061
 - incidence matrix of, 448 pr., 593 ex.
 - interval, 422 ex.
 - nonhamiltonian, 1061
 - shortest path in, 597
 - singly connected, 612 ex.
 - sparse, 589
 - static, 562 n.
 - subproblem, 367–368
 - tour of, 1096
 - weighted, 591
 - see also* directed acyclic graph, directed graph, flow network, undirected graph, tree
- graphic matroid, 437–438, 642
- GRAPH-ISOMORPHISM, 1065 ex.
- gray vertex, 594, 603
- greatest common divisor (gcd), 929–930, 933 ex.
 - binary gcd algorithm for, 981 pr.
 - Euclid's algorithm for, 933–939, 981 pr., 983
 - with more than two arguments, 939 ex.
 - recursion theorem for, 934
- greedoid, 450
- GREEDY, 440
 - GREEDY-ACTIVITY-SELECTOR, 421
 - greedy algorithm, 414–450
 - for activity selection, 415–422
 - for coin changing, 446 pr.
 - compared with dynamic programming, 381, 390 ex., 418, 423–427
 - Dijkstra's algorithm, 658–664
 - elements of, 423–428
 - for fractional knapsack problem, 426
 - greedy-choice property in, 424–425
 - for Huffman code, 428–437
 - Kruskal's algorithm, 631–633
 - and matroids, 437–443
 - for minimum spanning tree, 631–638
 - for multithreaded scheduling, 781–783
 - for off-line caching, 449 pr.
 - optimal substructure in, 425
 - Prim's algorithm, 634–636
 - for set cover, 1117–1122, 1139
 - for task scheduling, 443–446, 447–448 pr.
 - on a weighted matroid, 439–442
 - for weighted set cover, 1135 pr.
 - greedy-choice property, 424–425
 - of activity selection, 417–418
 - of Huffman codes, 433–434
 - of a weighted matroid, 441
 - greedy scheduler, 782
 - GREEDY-SET-COVER, 1119
 - grid, 760 pr.
 - group, 939–946
 - cyclic, 955
 - operator (\oplus), 939
 - guessing the solution, in the substitution method, 84–85
 - half 3-CNF satisfiability, 1101 ex.
 - half-open interval, 348
 - Hall's theorem, 735 ex.
 - halting problem, 1048
 - halving lemma, 908
 - HAM-CYCLE, 1062
 - hamiltonian cycle, 1049, 1061, 1091–1096, 1105
 - hamiltonian graph, 1061
 - hamiltonian path, 1066 ex., 1101 ex.
 - HAM-PATH, 1066 ex.
 - handle, 163, 507
 - handshaking lemma, 1172 ex.

- harmonic number, 1147, 1153–1154
- harmonic series, 1147, 1153–1154
- HASH-DELETE, 277 ex.
- hash function, 256, 262–269
 - auxiliary, 272
 - collision-resistant, 964
 - division method for, 263, 268–269 ex.
 - ϵ -universal, 269 ex.
 - multiplication method for, 263–264
 - universal, 265–268
- hashing, 253–285
 - with chaining, 257–260, 283 pr.
 - double, 272–274, 277 ex.
 - k -universal, 284 pr.
 - in memoization, 365, 387
 - with open addressing, 269–277
 - perfect, 277–282, 285
 - to replace adjacency lists, 593 ex.
 - universal, 265–268
- HASH-INSERT, 270, 277 ex.
- HASH-SEARCH, 271, 277 ex.
- hash table, 256–261
 - dynamic, 471 ex.
 - secondary, 278
 - see also* hashing
- hash value, 256
- hat-check problem, 122 ex.
- head
 - in a disk drive, 485
 - of a linked list, 236
 - of a queue, 234
- heap, 151–169
 - analyzed by potential method, 462 ex.
 - binomial, 527 pr.
 - building, 156–159, 166 pr.
 - compared with Fibonacci heaps, 506–507
 - d -ary, 167 pr., 706 pr.
 - deletion from, 166 ex.
 - in Dijkstra's algorithm, 662
 - extracting the maximum key from, 163
 - Fibonacci, *see* Fibonacci heap
 - as garbage-collected storage, 151
 - height of, 153
 - in Huffman's algorithm, 433
 - to implement a mergeable heap, 506
 - increasing a key in, 163–164
 - insertion into, 164
 - in Johnson's algorithm, 704
 - max-heap, 152
 - maximum key of, 163
 - mergeable, *see* mergeable heap
 - min-heap, 153
 - in Prim's algorithm, 636
 - as a priority queue, 162–166
 - relaxed, 530
 - running times of operations on, 506 fig.
 - and treaps, 333 pr.
 - 2-3-4, 529 pr.
- HEAP-DECREASE-KEY, 165 ex.
- HEAP-DELETE, 166 ex.
- HEAP-EXTRACT-MAX, 163
- HEAP-EXTRACT-MIN, 165 ex.
- HEAP-INCREASE-KEY, 164
- HEAP-MAXIMUM, 163
- HEAP-MINIMUM, 165 ex.
- heap property, 152
 - maintenance of, 154–156
 - vs. binary-search-tree property, 289 ex.
- heapsort, 151–169
- HEAPSORT, 160
- heel, 602 ex.
- height
 - of a binomial tree, 527 pr.
 - black-, 309
 - of a B-tree, 489–490
 - of a d -ary heap, 167 pr.
 - of a decision tree, 193
 - exponential, 300
 - of a heap, 153
 - of a node in a heap, 153, 159 ex.
 - of a node in a tree, 1177
 - of a red-black tree, 309
 - of a tree, 1177
- height-balanced tree, 333 pr.
- height function, in push-relabel algorithms, 738
- hereditary family of subsets, 437
- Hermitian matrix, 832 ex.
- high endpoint of an interval, 348
- high function, 537, 546
- HIRE-ASSISTANT, 115
- hiring problem, 114–115, 123–124, 145
 - on-line, 139–141
 - probabilistic analysis of, 120–121
- hit
 - cache, 449 pr.
 - spurious, 991

- HOARE-PARTITION, 185 pr.
- HOPCROFT-KARP, 764 pr.
- Hopcroft-Karp bipartite matching algorithm, 763 pr.
- horizontal ray, 1021 ex.
- Horner's rule, 41 pr., 900
 - in the Rabin-Karp algorithm, 990
- HUFFMAN, 431
- Huffman code, 428–437, 450
- hull, convex, 8, 1029–1039, 1046 pr.
- Human Genome Project, 6
- hyperedge, 1172
- hypergraph, 1172
 - and bipartite graphs, 1173 ex.
- ideal parallel computer, 779
- idempotency laws for sets, 1159
- identity, 939
- identity matrix, 1218
- if**, in pseudocode, 20
- image, 1167
- image compression, 409 pr., 413
- inadmissible edge, 749
- incidence, 1169
- incidence matrix
 - and difference constraints, 666
 - of a directed graph, 448 pr., 593 ex.
 - of an undirected graph, 448 pr.
- inclusion and exclusion, 1163 ex.
- incomplete step, 782
- INCREASE-KEY, 162
- increasing a key, in a max-heap, 163–164
- INCREMENT, 454
- incremental design method, 29
 - for finding the convex hull, 1030
- in-degree, 1169
- indentation in pseudocode, 20
- independence
 - of events, 1192–1193, 1195 ex.
 - of random variables, 1197
 - of subproblems in dynamic programming, 383–384
- independent family of subsets, 437
- independent set, 1101 pr.
 - of tasks, 444
- independent strands, 789
- index function, 537, 546
- index of an element of \mathbb{Z}_n^* , 955
- indicator random variable, 118–121
 - in analysis of expected height of a randomly built binary search tree, 300–303
 - in analysis of inserting into a treap, 333 pr.
 - in analysis of streaks, 138–139
 - in analysis of the birthday paradox, 132–133
 - in approximation algorithm for MAX-3-CNF satisfiability, 1124
 - in bounding the right tail of the binomial distribution, 1212–1213
 - in bucket sort analysis, 202–204
 - expected value of, 118
 - in hashing analysis, 259–260
 - in hiring-problem analysis, 120–121
 - and linearity of expectation, 119
 - in quicksort analysis, 182–184, 187 pr.
 - in randomized-selection analysis, 217–219, 226 pr.
 - in universal-hashing analysis, 265–266
- induced subgraph, 1171
- inequality constraint, 852
 - and equality constraints, 853
- inequality, linear, 846
- infeasible linear program, 851
- infeasible solution, 851
- infinite sequence, 1166
- infinite set, 1161
- infinite sum, 1145
- infinity, arithmetic with, 650
- INITIALIZE-PREFLOW, 740
- INITIALIZE-SIMPLEX, 871, 887
- INITIALIZE-SINGLE-SOURCE, 648
- initial strand, 779
- injective function, 1167
- inner product, 1222
- inorder tree walk, 287, 293 ex., 342
- INORDER-TREE-WALK, 288
- in-place sorting, 17, 148, 206 pr.
- input
 - to an algorithm, 5
 - to a combinational circuit, 1071
 - distribution of, 116, 122
 - to a logic gate, 1070
 - size of, 25
- input alphabet, 995
- INSERT, 162, 230, 463 ex., 505
- insertion
 - into binary search trees, 294–295

- into a bit vector with a superimposed binary tree, 534
- into a bit vector with a superimposed tree of constant height, 534
- into B-trees, 493–497
- into chained hash tables, 258
- into d -ary heaps, 167 pr.
- into direct-address tables, 254
- into dynamic tables, 464–467
- elementary, 465
- into Fibonacci heaps, 510–511
- into heaps, 164
- into interval trees, 349
- into linked lists, 237–238
- into open-address hash tables, 270
- into order-statistic trees, 343
- into proto van Emde Boas structures, 544
- into queues, 234
- into red-black trees, 315–323
- into stacks, 232
- into sweep-line statuses, 1024
- into treaps, 333 pr.
- into 2-3-4 heaps, 529 pr.
- into van Emde Boas trees, 552–554
- into Young tableaux, 167 pr.
- insertion sort, 12, 16–20, 25–27
 - in bucket sort, 201–204
 - compared with merge sort, 14 ex.
 - compared with quicksort, 178 ex.
 - decision tree for, 192 fig.
 - in merge sort, 39 pr.
 - in quicksort, 185 ex.
 - using binary search, 39 ex.
- INSERTION-SORT, 18, 26, 208 pr.
- instance
 - of an abstract problem, 1051, 1054
 - of a problem, 5
- instructions of the RAM model, 23
- integer data type, 23
- integer linear programming, 850, 895 pr., 1101 ex.
- integers (\mathbb{Z}), 1158
- integer-valued flow, 733
- integrality theorem, 734
- integral, to approximate summations, 1154–1156
- integration of a series, 1147
- interior of a polygon, 1020 ex.
- interior-point method, 850, 897
- intermediate vertex, 693
- internal node, 1176
- internal path length, 1180 ex.
- interpolation by a cubic spline, 840 pr.
- interpolation by a polynomial, 901, 906 ex.
 - at complex roots of unity, 912–913
- intersection
 - of chords, 345 ex.
 - determining, for a set of line segments, 1021–1029, 1047
 - determining, for two line segments, 1017–1019
 - of languages, 1058
 - of sets (\cap), 1159
- interval, 348
 - fuzzy sorting of, 189 pr.
- INTERVAL-DELETE, 349
- interval graph, 422 ex.
- INTERVAL-INSERT, 349
- INTERVAL-SEARCH, 349, 351
- INTERVAL-SEARCH-EXACTLY, 354 ex.
- interval tree, 348–354
- interval trichotomy, 348
- intractability, 1048
- invalid shift, 985
- inventory planning, 411 pr.
- inverse
 - of a bijective function, 1167
 - in a group, 940
 - of a matrix, 827–831, 842, 1223, 1225 ex.
 - multiplicative, modulo n , 949
- inversion
 - in a self-organizing list, 476 pr.
 - in a sequence, 41 pr., 122 ex., 345 ex.
- inverter, 1070
- invertible matrix, 1223
- isolated vertex, 1169
- isomorphic graphs, 1171
- iterated function, 63 pr.
- iterated logarithm function, 58–59
- ITERATIVE-FFT, 917
- ITERATIVE-TREE-SEARCH, 291
- iter function, 577
- Jarvis's march, 1037–1038, 1047
- Jensen's inequality, 1199
- JOHNSON, 704

- Johnson's algorithm, 700–706
- joining
 - of red-black trees, 332 pr.
 - of 2-3-4 trees, 503 pr.
- joint probability density function, 1197
- Josephus permutation, 355 pr.
- Karmarkar's algorithm, 897
- Karp's minimum mean-weight cycle algorithm, 680 pr.
- k -ary tree, 1179
- k -CNF, 1049
- k -coloring, 1103 pr., 1180 pr.
- k -combination, 1185
- k -conjunctive normal form, 1049
- kernel of a polygon, 1038 ex.
- key, 16, 147, 162, 229
 - dummy, 397
 - interpreted as a natural number, 263
 - median, of a B-tree node, 493
 - public, 959, 962
 - secret, 959, 962
 - static, 277
- keywords, in pseudocode, 20–22
 - multithreaded, 774, 776–777, 785–786
- "killer adversary" for quicksort, 190
- Kirchhoff's current law, 708
- Kleene star (*), 1058
- KMP algorithm, 1002–1013
- KMP-MATCHER, 1005
- knapsack problem
 - fractional, 426, 428 ex.
 - 0-1, 425, 427 ex., 1137 pr., 1139
- k -neighbor tree, 338
- knot, of a spline, 840 pr.
- Knuth-Morris-Pratt algorithm, 1002–1013
- k -permutation, 126, 1184
- Kraft inequality, 1180 ex.
- Kruskal's algorithm, 631–633, 642
 - with integer edge weights, 637 ex.
- k -sorted, 207 pr.
- k -string, 1184
- k -subset, 1161
- k -substring, 1184
- k th power, 933 ex.
- k -universal hashing, 284 pr.
- Lagrange's formula, 902
- Lagrange's theorem, 944
- Lamé's theorem, 936
- language, 1057
 - completeness of, 1077 ex.
 - proving NP-completeness of, 1078–1079
 - verification of, 1063
- last-in, first-out, 232
 - see also* stack
- late task, 444
- layers
 - convex, 1044 pr.
 - maximal, 1045 pr.
- LCA, 584 pr.
- lcm (least common multiple), 939 ex.
- LCS, 7, 390–397, 413
- LCS-LENGTH, 394
- leading submatrix, 833, 839 ex.
- leaf, 1176
- least common ancestor, 584 pr.
- least common multiple, 939 ex.
- least-squares approximation, 835–839
- leaving a vertex, 1169
- leaving variable, 867
- LEFT, 152
- left child, 1178
- left-child, right-sibling representation, 246, 249 ex.
- LEFT-ROTATE, 313, 353 ex.
- left rotation, 312
- left spine, 333 pr.
- left subtree, 1178
- Legendre symbol ($\frac{a}{p}$), 982 pr.
- length
 - of a path, 1170
 - of a sequence, 1166
 - of a spine, 333 pr.
 - of a string, 986, 1184
- level
 - of a function, 573
 - of a tree, 1177
- level function, 576
- lexicographically less than, 304 pr.
- lexicographic sorting, 304 pr.
- lg (binary logarithm), 56
- lg* (iterated logarithm function), 58–59
- lg ^{k} (exponentiation of logarithms), 56
- lg lg (composition of logarithms), 56
- LIFO (last-in, first-out), 232

- see also* stack
- light edge, 626
- linear constraint, 846
- linear dependence, 1223
- linear equality, 845
- linear equations
 - solving modular, 946–950
 - solving systems of, 813–827
 - solving tridiagonal systems of, 840 pr.
- linear function, 26, 845
- linear independence, 1223
- linear inequality, 846
- linear-inequality feasibility problem, 894 pr.
- linearity of expectation, 1198
 - and indicator random variables, 119
- linearity of summations, 1146
- linear order, 1165
- linear permutation, 1229 pr.
- linear probing, 272
- linear programming, 7, 843–897
 - algorithms for, 850
 - applications of, 849
 - duality in, 879–886
 - ellipsoid algorithm for, 850, 897
 - finding an initial solution in, 886–891
 - fundamental theorem of, 892
 - interior-point methods for, 850, 897
 - Karmarkar’s algorithm for, 897
 - and maximum flow, 860–861
 - and minimum-cost circulation, 896 pr.
 - and minimum-cost flow, 861–862
 - and minimum-cost multicommodity flow, 864 ex.
 - and multicommodity flow, 862–863
 - simplex algorithm for, 864–879, 896
 - and single-pair shortest path, 859–860
 - and single-source shortest paths, 664–670, 863 ex.
 - slack form for, 854–857
 - standard form for, 850–854
 - see also* integer linear programming, 0-1 integer programming
- linear-programming relaxation, 1125
- linear search, 22 ex.
- linear speedup, 780
- line segment, 1015
 - comparable, 1022
 - determining turn of, 1017
 - determining whether any intersect, 1021–1029, 1047
 - determining whether two intersect, 1017–1019
- link
 - of binomial trees, 527 pr.
 - of Fibonacci-heap roots, 513
 - of trees in a disjoint-set forest, 570–571
- LINK, 571
- linked list, 236–241
 - compact, 245 ex., 250 pr.
 - deletion from, 238
 - to implement disjoint sets, 564–568
 - insertion into, 237–238
 - neighbor list, 750
 - searching, 237, 268 ex.
 - self-organizing, 476 pr.
- list, *see* linked list
- LIST-DELETE, 238
- LIST-DELETE’, 238
- LIST-INSERT, 238
- LIST-INSERT’, 240
- LIST-SEARCH, 237
- LIST-SEARCH’, 239
- literal, 1082
- little-oh notation, 50–51, 64
- little-omega notation, 51
- L_m -distance, 1044 ex.
- ln (natural logarithm), 56
- load factor
 - of a dynamic table, 463
 - of a hash table, 258
- load instruction, 23
- local variable, 21
- logarithm function (log), 56–57
 - discrete, 955
 - iterated (\lg^*), 58–59
- logical parallelism, 777
- logic gate, 1070
- longest common subsequence, 7, 390–397, 413
- longest palindrome subsequence, 405 pr.
- LONGEST-PATH, 1060 ex.
- LONGEST-PATH-LENGTH, 1060 ex.
- longest simple cycle, 1101 ex.
- longest simple path, 1048
 - in an unweighted graph, 382
 - in a weighted directed acyclic graph, 404 pr.
- LOOKUP-CHAIN, 388

- loop, in pseudocode, 20
 - parallel, 785–787
- loop invariant, 18–19
 - for breadth-first search, 595
 - for building a heap, 157
 - for consolidating the root list of a Fibonacci heap, 517
 - for determining the rank of an element in an order-statistic tree, 342
 - for Dijkstra’s algorithm, 660
 - and **for** loops, 19 n.
 - for the generic minimum-spanning-tree method, 625
 - for the generic push-relabel algorithm, 743
 - for Graham’s scan, 1034
 - for heapsort, 160 ex.
 - for Horner’s rule, 41 pr.
 - for increasing a key in a heap, 166 ex.
 - initialization of, 19
 - for insertion sort, 18
 - maintenance of, 19
 - for merging, 32
 - for modular exponentiation, 957
 - origin of, 42
 - for partitioning, 171
 - for Prim’s algorithm, 636
 - for the Rabin-Karp algorithm, 993
 - for randomly permuting an array, 127, 128 ex.
 - for red-black tree insertion, 318
 - for the relabel-to-front algorithm, 755
 - for searching an interval tree, 352
 - for the simplex algorithm, 872
 - for string-matching automata, 998, 1000
 - and termination, 19
- low endpoint of an interval, 348
- lower bounds
 - on approximations, 1140
 - asymptotic, 48
 - for average sorting, 207 pr.
 - on binomial coefficients, 1186
 - for comparting water jugs, 206 pr.
 - for convex hull, 1038 ex., 1047
 - for disjoint-set data structures, 585
 - for finding the minimum, 214
 - for finding the predecessor, 560
 - for length of an optimal traveling-salesman tour, 1112–1115
 - for median finding, 227
 - for merging, 208 pr.
 - for minimum-weight vertex cover, 1124–1126
 - for multithreaded computations, 780
 - and potential functions, 478
 - for priority-queue operations, 531
 - and recurrences, 67
 - for simultaneous minimum and maximum, 215 ex.
 - for size of an optimal vertex cover, 1110, 1135 pr.
 - for sorting, 191–194, 205 pr., 211, 531
 - for streaks, 136–138, 142 ex.
 - on summations, 1152, 1154
- lower median, 213
- lower square root ($\sqrt[n]{}$), 546
- lower-triangular matrix, 1219, 1222 ex., 1225 ex.
- low function, 537, 546
- LU decomposition, 806 pr., 819–822
- LU-DECOMPOSITION, 821
- LUP decomposition, 806 pr., 815
 - computation of, 823–825
 - of a diagonal matrix, 827 ex.
 - in matrix inversion, 828
 - and matrix multiplication, 832 ex.
 - of a permutation matrix, 827 ex.
 - use of, 815–819
- LUP-DECOMPOSITION, 824
- LUP-SOLVE, 817
- main memory, 484
- MAKE-HEAP, 505
- MAKE-SET, 561
 - disjoint-set-forest implementation of, 571
 - linked-list implementation of, 564
- makespan, 1136 pr.
- MAKE-TREE, 583 pr.
- Manhattan distance, 225 pr., 1044 ex.
- marked node, 508, 519–520
- Markov’s inequality, 1201 ex.
- master method for solving a recurrence, 93–97
- master theorem, 94
 - proof of, 97–106
- matched vertex, 732
- matching
 - bipartite, 732, 763 pr.

- maximal, 1110, 1135 pr.
- maximum, 1135 pr.
- and maximum flow, 732–736, 747 ex.
- perfect, 735 ex.
- of strings, 985–1013
- weighted bipartite, 530
- matric matroid, 437
- matrix, 1217–1229
 - addition of, 1220
 - adjacency, 591
 - conjugate transpose of, 832 ex.
 - determinant of, 1224–1225
 - diagonal, 1218
 - Hermitian, 832 ex.
 - identity, 1218
 - incidence, 448 pr., 593 ex.
 - inversion of, 806 pr., 827–831, 842
 - lower-triangular, 1219, 1222 ex., 1225 ex.
 - multiplication of, *see* matrix multiplication
 - negative of, 1220
 - permutation, 1220, 1222 ex.
 - predecessor, 685
 - product of, with a vector, 785–787, 789–790, 792 ex.
 - pseudoinverse of, 837
 - scalar multiple of, 1220
 - subtraction of, 1221
 - symmetric, 1220
 - symmetric positive-definite, 832–835, 842
 - Toeplitz, 921 pr.
 - transpose of, 797 ex., 1217
 - transpose of, multithreaded, 792 ex.
 - tridiagonal, 1219
 - unit lower-triangular, 1219
 - unit upper-triangular, 1219
 - upper-triangular, 1219, 1225 ex.
 - Vandermonde, 902, 1226 pr.
- matrix-chain multiplication, 370–378
- MATRIX-CHAIN-MULTIPLY
- MATRIX-CHAIN-ORDER, 375
- matrix multiplication, 75–83, 1221
 - for all-pairs shortest paths, 686–693, 706–707
 - boolean, 832 ex.
 - and computing the determinant, 832 ex.
 - divide-and-conquer method for, 76–83
 - and LUP decomposition, 832 ex.
 - and matrix inversion, 828–831, 842
 - multithreaded algorithm for, 792–797, 806 pr.
 - Pan’s method for, 82 ex.
 - Strassen’s algorithm for, 79–83, 111–112
- MATRIX-MULTIPLY, 371
- matrix-vector multiplication, multithreaded, 785–787, 792 ex.
 - with race, 789–790
- matroid, 437–443, 448 pr., 450, 642
 - for task scheduling, 443–446
- MAT-VEC, 785
- MAT-VEC-MAIN-LOOP, 786
- MAT-VEC-WRONG, 790
- MAX-CNF satisfiability, 1127 ex.
- MAX-CUT problem, 1127 ex.
- MAX-FLOW-BY-SCALING, 763 pr.
- max-flow min-cut theorem, 723
- max-heap, 152
 - building, 156–159
 - d -ary, 167 pr.
 - deletion from, 166 ex.
 - extracting the maximum key from, 163
 - in heapsort, 159–162
 - increasing a key in, 163–164
 - insertion into, 164
 - maximum key of, 163
 - as a max-priority queue, 162–166
 - mergeable, 250 n., 481 n., 505 n.
- MAX-HEAPIFY, 154
- MAX-HEAP-INSERT, 164
 - building a heap with, 166 pr.
- max-heap property, 152
 - maintenance of, 154–156
- maximal element, of a partially ordered set, 1165
- maximal layers, 1045 pr.
- maximal matching, 1110, 1135 pr.
- maximal point, 1045 pr.
- maximal subset, in a matroid, 438
- maximization linear program, 846
 - and minimization linear programs, 852
- maximum, 213
 - in binary search trees, 291
 - of a binomial distribution, 1207 ex.
 - in a bit vector with a superimposed binary tree, 533
 - in a bit vector with a superimposed tree of constant height, 535

- finding, 214–215
- in heaps, 163
- in order-statistic trees, 347 ex.
- in proto van Emde Boas structures, 544 ex.
- in red-black trees, 311
- in van Emde Boas trees, 550
- MAXIMUM, 162–163, 230
- maximum bipartite matching, 732–736, 747 ex., 766
 - Hopcroft-Karp algorithm for, 763 pr.
- maximum degree, in a Fibonacci heap, 509, 523–526
- maximum flow, 708–766
 - Edmonds-Karp algorithm for, 727–730
 - Ford-Fulkerson method for, 714–731, 765
 - as a linear program, 860–861
 - and maximum bipartite matching, 732–736, 747 ex.
 - push-relabel algorithms for, 736–760, 765
 - relabel-to-front algorithm for, 748–760
 - scaling algorithm for, 762 pr., 765
 - updating, 762 pr.
- maximum matching, 1135 pr.
- maximum spanning tree, 1137 pr.
- maximum-subarray problem, 68–75, 111
- max-priority queue, 162
- MAX-3-CNF satisfiability, 1123–1124, 1139
- MAYBE-MST-A, 641 pr.
- MAYBE-MST-B, 641 pr.
- MAYBE-MST-C, 641 pr.
- mean, *see* expected value
- mean weight of a cycle, 680 pr.
- median, 213–227
 - multithreaded algorithm for, 805 ex.
 - of sorted lists, 223 ex.
 - of two sorted lists, 804 ex.
 - weighted, 225 pr.
- median key, of a B-tree node, 493
- median-of-3 method, 188 pr.
- member of a set (\in), 1158
- membership
 - in proto van Emde Boas structures, 540–541
 - in Van Emde Boas trees, 550
- memoization, 365, 387–389
- MEMOIZED-CUT-ROD, 365
- MEMOIZED-CUT-ROD-AUX, 366
- MEMOIZED-MATRIX-CHAIN, 388
- memory, 484
- memory hierarchy, 24
- MERGE, 31
- mergeable heap, 481, 505
 - binomial heaps, 527 pr.
 - linked-list implementation of, 250 pr.
 - relaxed heaps, 530
 - running times of operations on, 506 fig.
 - 2-3-4 heaps, 529 pr.
 - see also* Fibonacci heap
- mergeable max-heap, 250 n., 481 n., 505 n.
- mergeable min-heap, 250 n., 481 n., 505
- MERGE-LISTS, 1129
- merge sort, 12, 30–37
 - compared with insertion sort, 14 ex.
 - multithreaded algorithm for, 797–805, 812
 - use of insertion sort in, 39 pr.
- MERGE-SORT, 34
- MERGE-SORT', 797
- merging
 - of k sorted lists, 166 ex.
 - lower bounds for, 208 pr.
 - multithreaded algorithm for, 798–801
 - of two sorted arrays, 30
- MILLER-RABIN, 970
- Miller-Rabin primality test, 968–975, 983
- MIN-GAP, 354 ex.
- min-heap, 153
 - analyzed by potential method, 462 ex.
 - building, 156–159
 - d -ary, 706 pr.
 - in Dijkstra's algorithm, 662
 - in Huffman's algorithm, 433
 - in Johnson's algorithm, 704
 - mergeable, 250 n., 481 n., 505
 - as a min-priority queue, 165 ex.
 - in Prim's algorithm, 636
- MIN-HEAPIFY, 156 ex.
- MIN-HEAP-INSERT, 165 ex.
- min-heap ordering, 507
- min-heap property, 153, 507
 - maintenance of, 156 ex.
 - in treaps, 333 pr.
 - vs. binary-search-tree property, 289 ex.
- minimization linear program, 846
 - and maximization linear programs, 852
- minimum, 213
 - in binary search trees, 291

- in a bit vector with a superimposed binary tree, 533
- in a bit vector with a superimposed tree of constant height, 535
- in B-trees, 497 ex.
- in Fibonacci heaps, 511
- finding, 214–215
- off-line, 582 pr.
- in order-statistic trees, 347 ex.
- in proto van Emde Boas structures, 541–542
- in red-black trees, 311
- in 2-3-4 heaps, 529 pr.
- in van Emde Boas trees, 550
- MINIMUM, 162, 214, 230, 505
- minimum-cost circulation, 896 pr.
- minimum-cost flow, 861–862
- minimum-cost multicommodity flow, 864 ex.
- minimum-cost spanning tree, *see* minimum spanning tree
- minimum cut, 721, 731 ex.
- minimum degree, of a B-tree, 489
- minimum mean-weight cycle, 680 pr.
- minimum node, of a Fibonacci heap, 508
- minimum path cover, 761 pr.
- minimum spanning tree, 624–642
 - in approximation algorithm for traveling-salesman problem, 1112
 - Borůvka's algorithm for, 641
 - on dynamic graphs, 637 ex.
 - generic method for, 625–630
 - Kruskal's algorithm for, 631–633
 - Prim's algorithm for, 634–636
 - relation to matroids, 437, 439–440
 - second-best, 638 pr.
- minimum-weight spanning tree, *see* minimum spanning tree
- minimum-weight vertex cover, 1124–1127, 1139
- minor of a matrix, 1224
- min-priority queue, 162
 - in constructing Huffman codes, 431
 - in Dijkstra's algorithm, 661
 - in Prim's algorithm, 634, 636
- miss, 449 pr.
- missing child, 1178
- mod, 54, 928
- modifying operation, 230
- modular arithmetic, 54, 923 pr., 939–946
- modular equivalence, 54, 1165 ex.
- modular exponentiation, 956
- MODULAR-EXPONENTIATION, 957
- modular linear equations, 946–950
- MODULAR-LINEAR-EQUATION-SOLVER, 949
- modulo, 54, 928
- Monge array, 110 pr.
- monotone sequence, 168
- monotonically decreasing, 53
- monotonically increasing, 53
- Monty Hall problem, 1195 ex.
- move-to-front heuristic, 476 pr., 478
- MST-KRUSKAL, 631
- MST-PRIM, 634
- MST-REDUCE, 639 pr.
- much-greater-than (\gg), 574
- much-less-than (\ll), 783
- multicommodity flow, 862–863
 - minimum-cost, 864 ex.
- multicore computer, 772
- multidimensional fast Fourier transform, 921 pr.
- multigraph, 1172
 - converting to equivalent undirected graph, 593 ex.
- multiple, 927
 - of an element modulo n , 946–950
 - least common, 939 ex.
 - scalar, 1220
- multiple assignment, 21
- multiple sources and sinks, 712
- multiplication
 - of complex numbers, 83 ex.
 - divide-and-conquer method for, 920 pr.
 - of matrices, *see* matrix multiplication
 - of a matrix chain, 370–378
 - matrix-vector, multithreaded, 785–787, 789–790, 792 ex.
 - modulo n ($_n$), 940
 - of polynomials, 899
- multiplication method, 263–264
- multiplicative group modulo n , 941
- multiplicative inverse, modulo n , 949
- multiply instruction, 23
- MULTIPOP, 453
- multiprocessor, 772
- MULTIPUSH, 456 ex.

- multiset, 1158 n.
- multithreaded algorithm, 10, 772–812
 - for computing Fibonacci numbers, 774–780
 - for fast Fourier transform, 804 ex.
 - Floyd-Warshall algorithm, 797 ex.
 - for LU decomposition, 806 pr.
 - for LUP decomposition, 806 pr.
 - for matrix inversion, 806 pr.
 - for matrix multiplication, 792–797, 806 pr.
 - for matrix transpose, 792 ex., 797 ex.
 - for matrix-vector product, 785–787, 789–790, 792 ex.
 - for median, 805 ex.
 - for merge sorting, 797–805, 812
 - for merging, 798–801
 - for order statistics, 805 ex.
 - for partitioning, 804 ex.
 - for prefix computation, 807 pr.
 - for quicksort, 811 pr.
 - for reduction, 807 pr.
 - for a simple stencil calculation, 809 pr.
 - for solving systems of linear equations, 806 pr.
 - Strassen's algorithm, 795–796
- multithreaded composition, 784 fig.
- multithreaded computation, 777
- multithreaded scheduling, 781–783
- mutually exclusive events, 1190
- mutually independent events, 1193
- \mathbb{N} (set of natural numbers), 1158
- naive algorithm, for string matching, 988–990
- NAIVE-STRING-MATCHER, 988
- natural cubic spline, 840 pr.
- natural numbers (\mathbb{N}), 1158
 - keys interpreted as, 263
- negative of a matrix, 1220
- negative-weight cycle
 - and difference constraints, 667
 - and relaxation, 677 ex.
 - and shortest paths, 645, 653–654, 692 ex., 700 ex.
- negative-weight edges, 645–646
- neighbor, 1172
- neighborhood, 735 ex.
- neighbor list, 750
- nested parallelism, 776, 805 pr.
- nesting boxes, 678 pr.
- net flow across a cut, 720
- network
 - admissible, 749–750
 - flow, *see* flow network
 - residual, 715–719
 - for sorting, 811
- NEXT-TO-TOP, 1031
- NIL, 21
- node, 1176
 - see also* vertex
- nonbasic variable, 855
- nondeterministic multithreaded algorithm, 787
- nondeterministic polynomial time, 1064 n.
 - see also* NP
- nonhamiltonian graph, 1061
- noninstance, 1056 n.
- noninvertible matrix, 1223
- nonnegativity constraint, 851, 853
- nonoverlappable string pattern, 1002 ex.
- nonsaturating push, 739, 745
- nonsingular matrix, 1223
- nontrivial power, 933 ex.
- nontrivial square root of 1, modulo n , 956
- no-path property, 650, 672
- normal equation, 837
- norm of a vector, 1222
- NOT function (\neg), 1071
- not a set member (\notin), 1158
- not equivalent (\neq), 54
- NOT gate, 1070
- NP (complexity class), 1049, 1064, 1066 ex., 1105
- NPC (complexity class), 1050, 1069
- NP-complete, 1050, 1069
- NP-completeness, 9–10, 1048–1105
 - of the circuit-satisfiability problem, 1070–1077
 - of the clique problem, 1086–1089, 1105
 - of determining whether a boolean formula is a tautology, 1086 ex.
 - of the formula-satisfiability problem, 1079–1081, 1105
 - of the graph-coloring problem, 1103 pr.
 - of the half 3-CNF satisfiability problem, 1101 ex.
 - of the hamiltonian-cycle problem, 1091–1096, 1105
 - of the hamiltonian-path problem, 1101 ex.

- of the independent-set problem, 1101 pr.
 - of integer linear programming, 1101 ex.
 - of the longest-simple-cycle problem, 1101 ex.
 - proving, of a language, 1078–1079
 - of scheduling with profits and deadlines, 1104 pr.
 - of the set-covering problem, 1122 ex.
 - of the set-partition problem, 1101 ex.
 - of the subgraph-isomorphism problem, 1100 ex.
 - of the subset-sum problem, 1097–1100
 - of the 3-CNF-satisfiability problem, 1082–1085, 1105
 - of the traveling-salesman problem, 1096–1097
 - of the vertex-cover problem, 1089–1091, 1105
 - of 0-1 integer programming, 1100 ex.
- NP-hard, 1069
- n -set, 1161
- n -tuple, 1162
- null event, 1190
- null tree, 1178
- null vector, 1224
- number-field sieve, 984
- numerical stability, 813, 815, 842
- n -vector, 1218
- o -notation, 50–51, 64
- O -notation, 45 fig., 47–48, 64
- O' -notation, 62 pr.
- \widetilde{O} -notation, 62 pr.
- object, 21
 - allocation and freeing of, 243–244
 - array implementation of, 241–246
 - passing as parameter, 21
- objective function, 664, 847, 851
- objective value, 847, 851
- oblivious compare-exchange algorithm, 208 pr.
- occurrence of a pattern, 985
- OFF-LINE-MINIMUM, 583 pr.
- off-line problem
 - caching, 449 pr.
 - least common ancestors, 584 pr.
 - minimum, 582 pr.
- Omega-notation, 45 fig., 48–49, 64
- 1-approximation algorithm, 1107
- one-pass method, 585
- one-to-one correspondence, 1167
- one-to-one function, 1167
- on-line convex-hull problem, 1039 ex.
- on-line hiring problem, 139–141
- ON-LINE-MAXIMUM, 140
- on-line multithreaded scheduler, 781
- ON-SEGMENT, 1018
- onto function, 1167
- open-address hash table, 269–277
 - with double hashing, 272–274, 277 ex.
 - with linear probing, 272
 - with quadratic probing, 272, 283 pr.
- open interval, 348
- OpenMP, 774
- optimal binary search tree, 397–404, 413
- OPTIMAL-BST, 402
- optimal objective value, 851
- optimal solution, 851
- optimal subset, of a matroid, 439
- optimal substructure
 - of activity selection, 416
 - of binary search trees, 399–400
 - in dynamic programming, 379–384
 - of the fractional knapsack problem, 426
 - in greedy algorithms, 425
 - of Huffman codes, 435
 - of longest common subsequences, 392–393
 - of matrix-chain multiplication, 373
 - of rod-cutting, 362
 - of shortest paths, 644–645, 687, 693–694
 - of unweighted shortest paths, 382
 - of weighted matroids, 442
 - of the 0-1 knapsack problem, 426
- optimal vertex cover, 1108
- optimization problem, 359, 1050, 1054
 - approximation algorithms for, 10, 1106–1140
 - and decision problems, 1051
- OR function (\vee), 697, 1071
- order
 - of a group, 945
 - linear, 1165
 - partial, 1165
 - total, 1165
- ordered pair, 1161
- ordered tree, 1177
- order of growth, 28

- order statistics, 213–227
 - dynamic, 339–345
 - multithreaded algorithm for, 805 ex.
- order-statistic tree, 339–345
 - querying, 347 ex.
- OR gate, 1070
- origin, 1015
- or, in pseudocode, 22
- orthonormal, 842
- OS-KEY-RANK, 344 ex.
- OS-RANK, 342
- OS-SELECT, 341
- out-degree, 1169
- outer product, 1222
- output
 - of an algorithm, 5
 - of a combinational circuit, 1071
 - of a logic gate, 1070
- overdetermined system of linear equations, 814
- overflow
 - of a queue, 235
 - of a stack, 233
- overflowing vertex, 736
 - discharge of, 751
- overlapping intervals, 348
 - finding all, 354 ex.
 - point of maximum overlap, 354 pr.
- overlapping rectangles, 354 ex.
- overlapping subproblems, 384–386
- overlapping-suffix lemma, 987
- P (complexity class), 1049, 1055, 1059, 1061 ex., 1105
- package wrapping, 1037, 1047
- page on a disk, 486, 499 ex., 502 pr.
- pair, ordered, 1161
- pairwise disjoint sets, 1161
- pairwise independence, 1193
- pairwise relatively prime, 931
- palindrome, 405 pr.
- Pan's method for matrix multiplication, 82 ex.
- parallel algorithm, 10, 772
 - see also* multithreaded algorithm
- parallel computer, 772
 - ideal, 779
- parallel for**, in pseudocode, 785–786
- parallelism
 - logical, 777
 - of a multithreaded computation, 780
 - nested, 776
 - of a randomized multithreaded algorithm, 811 pr.
- parallel loop, 785–787, 805 pr.
- parallel-machine-scheduling problem, 1136 pr.
- parallel prefix, 807 pr.
- parallel random-access machine, 811
- parallel slackness, 781
 - rule of thumb, 783
- parallel, strands being logically in, 778
- parameter, 21
 - costs of passing, 107 pr.
- parent
 - in a breadth-first tree, 594
 - in a multithreaded computation, 776
 - in a rooted tree, 1176
- PARENT, 152
- parenthesis structure of depth-first search, 606
- parenthesis theorem, 606
- parenthesization of a matrix-chain product, 370
- parse tree, 1082
- partially ordered set, 1165
- partial order, 1165
- PARTITION, 171
- PARTITION', 186 pr.
- partition function, 361 n.
- partitioning, 171–173
 - around median of 3 elements, 185 ex.
 - Hoare's method for, 185 pr.
 - multithreaded algorithm for, 804 ex.
 - randomized, 179
- partition of a set, 1161, 1164
- Pascal's triangle, 1188 ex.
- path, 1170
 - augmenting, 719–720, 763 pr.
 - critical, 657
 - find, 569
 - hamiltonian, 1066 ex.
 - longest, 382, 1048
 - shortest, *see* shortest paths
 - simple, 1170
 - weight of, 643
- PATH, 1051, 1058
- path compression, 569
- path cover, 761 pr.
- path length, of a tree, 304 pr., 1180 ex.
- path-relaxation property, 650, 673

- pattern, in string matching, 985
 - nonoverlappable, 1002 ex.
- pattern matching, *see* string matching
- penalty, 444
- perfect hashing, 277–282, 285
- perfect linear speedup, 780
- perfect matching, 735 ex.
- permutation, 1167
 - bit-reversal, 472 pr., 918
 - Josephus, 355 pr.
 - k -permutation, 126, 1184
 - linear, 1229 pr.
 - in place, 126
 - random, 124–128
 - of a set, 1184
 - uniform random, 116, 125
- permutation matrix, 1220, 1222 ex., 1226 ex.
 - LUP decomposition of, 827 ex.
- PERMUTE-BY-CYCLIC, 129 ex.
- PERMUTE-BY-SORTING, 125
- PERMUTE-WITH-ALL, 129 ex.
- PERMUTE-WITHOUT-IDENTITY, 128 ex.
- persistent data structure, 331 pr., 482
- PERSISTENT-TREE-INSERT, 331 pr.
- PERT chart, 657, 657 ex.
- P-FIB, 776
- phase, of the relabel-to-front algorithm, 758
- phi function ($\phi(n)$), 943
- PISANO-DELETE, 526 pr.
- pivot
 - in linear programming, 867, 869–870, 878 ex.
 - in LU decomposition, 821
 - in quicksort, 171
- PIVOT, 869
- platter, 485
- P-MATRIX-MULTIPLY-RECURSIVE, 794
- P-MERGE, 800
- P-MERGE-SORT, 803
- pointer, 21
 - array implementation of, 241–246
 - trailing, 295
- point-value representation, 901
- polar angle, 1020 ex.
- Pollard's rho heuristic, 976–980, 980 ex., 984
- POLLARD-RHO, 976
- polygon, 1020 ex.
 - kernel of, 1038 ex.
 - star-shaped, 1038 ex.
- polylogarithmically bounded, 57
- polynomial, 55, 898
 - addition of, 898
 - asymptotic behavior of, 61 pr.
 - coefficient representation of, 900
 - derivatives of, 922 pr.
 - evaluation of, 41 pr., 900, 905 ex., 923 pr.
 - interpolation by, 901, 906 ex.
 - multiplication of, 899, 903–905, 920 pr.
 - point-value representation of, 901
- polynomial-growth condition, 113
- polynomially bounded, 55
- polynomially related, 1056
- polynomial-time acceptance, 1058
- polynomial-time algorithm, 927, 1048
- polynomial-time approximation scheme, 1107
 - for maximum clique, 1134 pr.
- polynomial-time computability, 1056
- polynomial-time decision, 1059
- polynomial-time reducibility (\leq_P), 1067, 1077 ex.
- polynomial-time solvability, 1055
- polynomial-time verification, 1061–1066
- POP, 233, 452
- pop from a run-time stack, 188 pr.
- positional tree, 1178
- positive-definite matrix, 1225
- post-office location problem, 225 pr.
- postorder tree walk, 287
- potential function, 459
 - for lower bounds, 478
- potential method, 459–463
 - for binary counters, 461–462
 - for disjoint-set data structures, 575–581, 582 ex.
 - for dynamic tables, 466–471
 - for Fibonacci heaps, 509–512, 517–518, 520–522
 - for the generic push-relabel algorithm, 746
 - for min-heaps, 462 ex.
 - for restructuring red-black trees, 474 pr.
 - for self-organizing lists with move-to-front, 476 pr.
 - for stack operations, 460–461
- potential, of a data structure, 459
- power
 - of an element, modulo n , 954–958

- k th, 933 ex.
- nontrivial, 933 ex.
- power series, 108 pr.
- power set, 1161
- $\Pr \{ \}$ (probability distribution), 1190
- PRAM, 811
- predecessor
 - in binary search trees, 291–292
 - in a bit vector with a superimposed binary tree, 534
 - in a bit vector with a superimposed tree of constant height, 535
 - in breadth-first trees, 594
 - in B-trees, 497 ex.
 - in linked lists, 236
 - in order-statistic trees, 347 ex.
 - in proto van Emde Boas structures, 544 ex.
 - in red-black trees, 311
 - in shortest-paths trees, 647
 - in Van Emde Boas trees, 551–552
- PREDECESSOR, 230
- predecessor matrix, 685
- predecessor subgraph
 - in all-pairs shortest paths, 685
 - in breadth-first search, 600
 - in depth-first search, 603
 - in single-source shortest paths, 647
- predecessor-subgraph property, 650, 676
- preemption, 447 pr.
- prefix
 - of a sequence, 392
 - of a string (\square), 986
- prefix code, 429
- prefix computation, 807 pr.
- prefix function, 1003–1004
- prefix-function iteration lemma, 1007
- preflow, 736, 765
- preimage of a matrix, 1228 pr.
- preorder, total, 1165
- preorder tree walk, 287
- presorting, 1043
- Prim's algorithm, 634–636, 642
 - with an adjacency matrix, 637 ex.
 - in approximation algorithm for traveling-salesman problem, 1112
 - implemented with a Fibonacci heap, 636
 - implemented with a min-heap, 636
 - with integer edge weights, 637 ex.
 - similarity to Dijkstra's algorithm, 634, 662
 - for sparse graphs, 638 pr.
- primality testing, 965–975, 983
 - Miller-Rabin test, 968–975, 983
 - pseudoprimalty testing, 966–968
- primal linear program, 880
- primary clustering, 272
- primary memory, 484
- prime distribution function, 965
- prime number, 928
 - density of, 965–966
- prime number theorem, 965
- primitive root of \mathbb{Z}_n^* , 955
- principal root of unity, 907
- principle of inclusion and exclusion, 1163 ex.
- PRINT-ALL-PAIRS-SHORTEST-PATH, 685
- PRINT-CUT-ROD-SOLUTION, 369
- PRINT-INTERSECTING-SEGMENTS, 1028 ex.
- PRINT-LCS, 395
- PRINT-OPTIMAL-PARENS, 377
- PRINT-PATH, 601
- PRINT-SET, 572 ex.
- priority queue, 162–166
 - in constructing Huffman codes, 431
 - in Dijkstra's algorithm, 661
 - heap implementation of, 162–166
 - lower bounds for, 531
 - max-priority queue, 162
 - min-priority queue, 162, 165 ex.
 - with monotone extractions, 168
 - in Prim's algorithm, 634, 636
 - proto van Emde Boas structure
 - implementation of, 538–545
 - van Emde Boas tree implementation of, 531–560
 - see also* binary search tree, binomial heap, Fibonacci heap
- probabilistically checkable proof, 1105, 1140
- probabilistic analysis, 115–116, 130–142
 - of approximation algorithm for MAX-3-CNF satisfiability, 1124
 - and average inputs, 28
 - of average node depth in a randomly built binary search tree, 304 pr.
 - of balls and bins, 133–134
 - of birthday paradox, 130–133
 - of bucket sort, 201–204, 204 ex.
 - of collisions, 261 ex., 282 ex.

- of convex hull over a sparse-hulled distribution, 1046 pr.
 - of file comparison, 995 ex.
 - of fuzzy sorting of intervals, 189 pr.
 - of hashing with chaining, 258–260
 - of height of a randomly built binary search tree, 299–303
 - of hiring problem, 120–121, 139–141
 - of insertion into a binary search tree with equal keys, 303 pr.
 - of longest-probe bound for hashing, 282 pr.
 - of lower bound for sorting, 205 pr.
 - of Miller-Rabin primality test, 971–975
 - and multithreaded algorithms, 811 pr.
 - of on-line hiring problem, 139–141
 - of open-address hashing, 274–276, 277 ex.
 - of partitioning, 179 ex., 185 ex., 187–188 pr.
 - of perfect hashing, 279–282
 - of Pollard’s rho heuristic, 977–980
 - of probabilistic counting, 143 pr.
 - of quicksort, 181–184, 187–188 pr., 303 ex.
 - of Rabin-Karp algorithm, 994
 - and randomized algorithms, 123–124
 - of randomized selection, 217–219, 226 pr.
 - of searching a compact list, 250 pr.
 - of slot-size bound for chaining, 283 pr.
 - of sorting points by distance from origin, 204 ex.
 - of streaks, 135–139
 - of universal hashing, 265–268
- probabilistic counting, 143 pr.
- probability, 1189–1196
- probability density function, 1196
- probability distribution, 1190
- probability distribution function, 204 ex.
- probe sequence, 270
- probing, 270, 282 pr.
 - see also* linear probing, quadratic probing, double hashing
- problem
 - abstract, 1054
 - computational, 5–6
 - concrete, 1055
 - decision, 1051, 1054
 - intractable, 1048
 - optimization, 359, 1050, 1054
 - solution to, 6, 1054–1055
 - tractable, 1048
- procedure, 6, 16–17
- product (\prod), 1148
 - Cartesian, 1162
 - cross, 1016
 - inner, 1222
 - of matrices, 1221, 1226 ex.
 - outer, 1222
 - of polynomials, 899
 - rule of, 1184
 - scalar flow, 714 ex.
- professional wrestler, 602 ex.
- program counter, 1073
- programming, *see* dynamic programming, linear programming
- proper ancestor, 1176
- proper descendant, 1176
- proper subgroup, 944
- proper subset (\subset), 1159
- proto van Emde Boas structure, 538–545
 - cluster in, 538
 - compared with van Emde Boas trees, 547
 - deletion from, 544
 - insertion into, 544
 - maximum in, 544 ex.
 - membership in, 540–541
 - minimum in, 541–542
 - predecessor in, 544 ex.
 - successor in, 543–544
 - summary in, 540
- PROTO-VEB-INSERT, 544
- PROTO-VEB-MEMBER, 541
- PROTO-VEB-MINIMUM, 542
- proto-vEB structure, *see* proto van Emde Boas structure
- PROTO-VEB-SUCCESSOR, 543
- prune-and-search method, 1030
- pruning a Fibonacci heap, 529 pr.
- P-SCAN-1, 808 pr.
- P-SCAN-2, 808 pr.
- P-SCAN-3, 809 pr.
- P-SCAN-DOWN, 809 pr.
- P-SCAN-UP, 809 pr.
- pseudocode, 16, 20–22
- pseudoinverse, 837
- pseudoprime, 966–968
- PSEUDOPRIME, 967
- pseudorandom-number generator, 117
- P-SQUARE-MATRIX-MULTIPLY, 793

- P-TRANPOSE, 792 ex.
- public key, 959, 962
- public-key cryptosystem, 958–965, 983
- PUSH
 - push-relabel operation, 739
 - stack operation, 233, 452
- push onto a run-time stack, 188 pr.
- push operation (in push-relabel algorithms), 738–739
 - nonsaturating, 739, 745
 - saturating, 739, 745
- push-relabel algorithm, 736–760, 765
 - basic operations in, 738–740
 - by discharging an overflowing vertex of maximum height, 760 ex.
 - to find a maximum bipartite matching, 747 ex.
 - gap heuristic for, 760 ex., 766
 - generic algorithm, 740–748
 - with a queue of overflowing vertices, 759 ex.
 - relabel-to-front algorithm, 748–760
- quadratic function, 27
- quadratic probing, 272, 283 pr.
- quadratic residue, 982 pr.
- quantile, 223 ex.
- query, 230
- queue, 232, 234–235
 - in breadth-first search, 595
 - implemented by stacks, 236 ex.
 - linked-list implementation of, 240 ex.
 - priority, *see* priority queue
 - in push-relabel algorithms, 759 ex.
- quicksort, 170–190
 - analysis of, 174–185
 - average-case analysis of, 181–184
 - compared with insertion sort, 178 ex.
 - compared with radix sort, 199
 - with equal element values, 186 pr.
 - good worst-case implementation of, 223 ex.
 - “killer adversary” for, 190
 - with median-of-3 method, 188 pr.
 - multithreaded algorithm for, 811 pr.
 - randomized version of, 179–180, 187 pr.
 - stack depth of, 188 pr.
 - tail-recursive version of, 188 pr.
 - use of insertion sort in, 185 ex.
 - worst-case analysis of, 180–181
- QUICKSORT, 171
- QUICKSORT', 186 pr.
- quotient, 928
- \mathbb{R} (set of real numbers), 1158
- Rabin-Karp algorithm, 990–995, 1013
- RABIN-KARP-MATCHER, 993
- race, 787–790
- RACE-EXAMPLE, 788
- radix sort, 197–200
 - compared with quicksort, 199
- RADIX-SORT, 198
- radix tree, 304 pr.
- RAM, 23–24
- RANDOM, 117
- random-access machine, 23–24
 - parallel, 811
- randomized algorithm, 116–117, 122–130
 - and average inputs, 28
 - comparison sort, 205 pr.
 - for fuzzy sorting of intervals, 189 pr.
 - for hiring problem, 123–124
 - for insertion into a binary search tree with equal keys, 303 pr.
 - for MAX-3-CNF satisfiability, 1123–1124, 1139
 - Miller-Rabin primality test, 968–975, 983
 - multithreaded, 811 pr.
 - for partitioning, 179, 185 ex., 187–188 pr.
 - for permuting an array, 124–128
 - Pollard's rho heuristic, 976–980, 980 ex., 984
 - and probabilistic analysis, 123–124
 - quicksort, 179–180, 185 ex., 187–188 pr.
 - randomized rounding, 1139
 - for searching a compact list, 250 pr.
 - for selection, 215–220
 - universal hashing, 265–268
 - worst-case performance of, 180 ex.
- RANDOMIZED-HIRE-ASSISTANT, 124
- RANDOMIZED-PARTITION, 179
- RANDOMIZED-QUICKSORT, 179, 303 ex.
 - relation to randomly built binary search trees, 304 pr.
- randomized rounding, 1139
- RANDOMIZED-SELECT, 216
- RANDOMIZE-IN-PLACE, 126

- randomly built binary search tree, 299–303, 304 pr.
- random-number generator, 117
- random permutation, 124–128
 - uniform, 116, 125
- RANDOM-SAMPLE, 130 ex.
- random sampling, 129 ex., 179
- RANDOM-SEARCH, 143 pr.
- random variable, 1196–1201
 - indicator, *see* indicator random variable
- range, 1167
 - of a matrix, 1228 pr.
- rank
 - column, 1223
 - full, 1223
 - of a matrix, 1223, 1226 ex.
 - of a node in a disjoint-set forest, 569, 575, 581 ex.
 - of a number in an ordered set, 300, 339
 - in order-statistic trees, 341–343, 344–345 ex.
 - row, 1223
- rate of growth, 28
- ray, 1021 ex.
- RB-DELETE, 324
- RB-DELETE-FIXUP, 326
- RB-ENUMERATE, 348 ex.
- RB-INSERT, 315
- RB-INSERT-FIXUP, 316
- RB-JOIN, 332 pr.
- RB-TRANSPLANT, 323
- reachability in a graph (\rightsquigarrow), 1170
- real numbers (\mathbb{R}), 1158
- reconstructing an optimal solution, in dynamic programming, 387
- record, 147
- rectangle, 354 ex.
- recurrence, 34, 65–67, 83–113
 - solution by Akra-Bazzi method, 112–113
 - solution by master method, 93–97
 - solution by recursion-tree method, 88–93
 - solution by substitution method, 83–88
- recurrence equation, *see* recurrence
- recursion, 30
- recursion tree, 37, 88–93
 - in proof of master theorem, 98–100
 - and the substitution method, 91–92
- RECURSIVE-ACTIVITY-SELECTOR, 419
- recursive case, 65
- RECURSIVE-FFT, 911
- RECURSIVE-MATRIX-CHAIN, 385
- red-black tree, 308–338
 - augmentation of, 346–347
 - compared with B-trees, 484, 490
 - deletion from, 323–330
 - in determining whether any line segments intersect, 1024
 - for enumerating keys in a range, 348 ex.
 - height of, 309
 - insertion into, 315–323
 - joining of, 332 pr.
 - maximum key of, 311
 - minimum key of, 311
 - predecessor in, 311
 - properties of, 308–312
 - relaxed, 311 ex.
 - restructuring, 474 pr.
 - rotation in, 312–314
 - searching in, 311
 - successor in, 311
 - see also* interval tree, order-statistic tree
- REDUCE, 807 pr.
- reduced-space van Emde Boas tree, 557 pr.
- reducibility, 1067–1068
- reduction algorithm, 1052, 1067
- reduction function, 1067
- reduction, of an array, 807 pr.
- reflexive relation, 1163
- reflexivity of asymptotic notation, 51
- region, feasible, 847
- regularity condition, 95
- rejection
 - by an algorithm, 1058
 - by a finite automaton, 996
- RELABEL, 740
- reabeled vertex, 740
- relabel operation, in push-relabel algorithms, 740, 745
- RELABEL-TO-FRONT, 755
- relabel-to-front algorithm, 748–760
 - phase of, 758
- relation, 1163–1166
- relatively prime, 931
- RELAX, 649
- relaxation
 - of an edge, 648–650
 - linear programming, 1125

- relaxed heap, 530
- relaxed red-black tree, 311 ex.
- release time, 447 pr.
- remainder, 54, 928
- remainder instruction, 23
- repeated squaring
 - for all-pairs shortest paths, 689–691
 - for raising a number to a power, 956
- repeat**, in pseudocode, 20
- repetition factor, of a string, 1012 pr.
- REPETITION-MATCHER, 1013 pr.
- representative of a set, 561
- RESET, 459 ex.
- residual capacity, 716, 719
- residual edge, 716
- residual network, 715–719
- residue, 54, 928, 982 pr.
- respecting a set of edges, 626
- return edge, 779
- return**, in pseudocode, 22
- return instruction, 23
- reweighting
 - in all-pairs shortest paths, 700–702
 - in single-source shortest paths, 679 pr.
- rho heuristic, 976–980, 980 ex., 984
- $\rho(n)$ -approximation algorithm, 1106, 1123
- RIGHT, 152
- right child, 1178
- right-conversion, 314 ex.
- right horizontal ray, 1021 ex.
- RIGHT-ROTATE, 313
- right rotation, 312
- right spine, 333 pr.
- right subtree, 1178
- rod-cutting, 360–370, 390 ex.
- root
 - of a tree, 1176
 - of unity, 906–907
 - of \mathbb{Z}_n^* , 955
- rooted tree, 1176
 - representation of, 246–249
- root list, of a Fibonacci heap, 509
- rotation
 - cyclic, 1012 ex.
 - in a red-black tree, 312–314
- rotational sweep, 1030–1038
- rounding, 1126
 - randomized, 1139
- row-major order, 394
- row rank, 1223
- row vector, 1218
- RSA public-key cryptosystem, 958–965, 983
- RS-vEB tree, 557 pr.
- rule of product, 1184
- rule of sum, 1183
- running time, 25
 - average-case, 28, 116
 - best-case, 29 ex., 49
 - expected, 28, 117
 - of a graph algorithm, 588
 - and multithreaded computation, 779–780
 - order of growth, 28
 - rate of growth, 28
 - worst-case, 27, 49
- sabermetrics, 412 n.
- safe edge, 626
- SAME-COMPONENT, 563
- sample space, 1189
- sampling, 129 ex., 179
- SAT, 1079
- satellite data, 147, 229
- satisfiability, 1072, 1079–1081, 1105, 1123–1124, 1127 ex., 1139
- satisfiable formula, 1049, 1079
- satisfying assignment, 1072, 1079
- saturated edge, 739
- saturating push, 739, 745
- scalar flow product, 714 ex.
- scalar multiple, 1220
- scaling
 - in maximum flow, 762 pr., 765
 - in single-source shortest paths, 679 pr.
- scan, 807 pr.
- SCAN, 807 pr.
- scapegoat tree, 338
- schedule, 444, 1136 pr.
 - event-point, 1023
- scheduler, for multithreaded computations, 777, 781–783, 812
 - centralized, 782
 - greedy, 782
 - work-stealing algorithm for, 812
- scheduling, 443–446, 447 pr., 450, 1104 pr., 1136 pr.
- Schur complement, 820, 834

- Schur complement lemma, 834
- SCRAMBLE-SEARCH, 143 pr.
- seam carving, 409 pr., 413
- SEARCH, 230
- searching, 22 ex.
 - binary search, 39 ex., 799–800
 - in binary search trees, 289–291
 - in B-trees, 491–492
 - in chained hash tables, 258
 - in compact lists, 250 pr.
 - in direct-address tables, 254
 - for an exact interval, 354 ex.
 - in interval trees, 350–353
 - linear search, 22 ex.
 - in linked lists, 237
 - in open-address hash tables, 270–271
 - in proto van Emde Boas structures, 540–541
 - in red-black trees, 311
 - in an unsorted array, 143 pr.
 - in Van Emde Boas trees, 550
- 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, 272
- secondary hash table, 278
- secondary storage
 - search tree for, 484–504
 - stacks on, 502 pr.
- second-best minimum spanning tree, 638 pr.
- secret key, 959, 962
- segment, *see* directed segment, line segment
- SEGMENTS-INTERSECT, 1018
- SELECT, 220
- selection, 213
 - of activities, 415–422, 450
 - and comparison sorts, 222
 - in expected linear time, 215–220
 - multithreaded, 805 ex.
 - in order-statistic trees, 340–341
 - in worst-case linear time, 220–224
- selection sort, 29 ex.
- selector vertex, 1093
- self-loop, 1168
- self-organizing list, 476 pr., 478
- semiconnected graph, 621 ex.
- sentinel, 31, 238–240, 309
- sequence $(())$
 - bitonic, 682 pr.
 - finite, 1166
 - infinite, 1166
 - inversion in, 41 pr., 122 ex., 345 ex.
 - probe, 270
- sequential consistency, 779, 812
- serial algorithm versus parallel algorithm, 772
- serialization, of a multithreaded algorithm, 774, 776
- series, 108 pr., 1146–1148
 - strands being logically in, 778
- set $(\{ \})$, 1158–1163
 - cardinality $(| |)$, 1161
 - convex, 714 ex.
 - difference $(-)$, 1159
 - independent, 1101 pr.
 - intersection (\cap) , 1159
 - member (\in) , 1158
 - not a member (\notin) , 1158
 - union (\cup) , 1159
- set-covering problem, 1117–1122, 1139
 - weighted, 1135 pr.
- set-partition problem, 1101 ex.
- shadow of a point, 1038 ex.
- shared memory, 772
- Shell's sort, 42
- shift, in string matching, 985
- shift instruction, 24
- short-circuiting operator, 22
- SHORTEST-PATH, 1050
- shortest paths, 7, 643–707
 - all-pairs, 644, 684–707
 - Bellman-Ford algorithm for, 651–655
 - with bitonic paths, 682 pr.
 - and breadth-first search, 597–600, 644
 - convergence property of, 650, 672–673
 - and difference constraints, 664–670
 - Dijkstra's algorithm for, 658–664
 - in a directed acyclic graph, 655–658
 - in ϵ -dense graphs, 706 pr.
 - estimate of, 648
 - Floyd-Warshall algorithm for, 693–697, 700 ex., 706
 - Gabow's scaling algorithm for, 679 pr.
 - Johnson's algorithm for, 700–706
 - as a linear program, 859–860
 - and longest paths, 1048

- by matrix multiplication, 686–693, 706–707
- and negative-weight cycles, 645, 653–654, 692 ex., 700 ex.
- with negative-weight edges, 645–646
- no-path property of, 650, 672
- optimal substructure of, 644–645, 687, 693–694
- path-relaxation property of, 650, 673
- predecessor-subgraph property of, 650, 676
- problem variants, 644
- and relaxation, 648–650
- by repeated squaring, 689–691
- single-destination, 644
- single-pair, 381, 644
- single-source, 643–683
- tree of, 647–648, 673–676
- triangle inequality of, 650, 671
- in an unweighted graph, 381, 597
- upper-bound property of, 650, 671–672
- in a weighted graph, 643
- sibling, 1176
- side of a polygon, 1020 ex.
- signature, 960
- simple cycle, 1170
- simple graph, 1170
- simple path, 1170
 - longest, 382, 1048
- simple polygon, 1020 ex.
- simple stencil calculation, 809 pr.
- simple uniform hashing, 259
- simplex, 848
- SIMPLEX, 871
- simplex algorithm, 848, 864–879, 896–897
- single-destination shortest paths, 644
- single-pair shortest path, 381, 644
 - as a linear program, 859–860
- single-source shortest paths, 643–683
 - Bellman-Ford algorithm for, 651–655
 - with bitonic paths, 682 pr.
 - and difference constraints, 664–670
 - Dijkstra’s algorithm for, 658–664
 - in a directed acyclic graph, 655–658
 - in ϵ -dense graphs, 706 pr.
 - Gabow’s scaling algorithm for, 679 pr.
 - as a linear program, 863 ex.
 - and longest paths, 1048
- singleton, 1161
- singly connected graph, 612 ex.
- singly linked list, 236
 - see also* linked list
- singular matrix, 1223
- singular value decomposition, 842
- sink vertex, 593 ex., 709, 712
- size
 - of an algorithm’s input, 25, 926–927, 1055–1057
 - of a binomial tree, 527 pr.
 - of a boolean combinational circuit, 1072
 - of a clique, 1086
 - of a set, 1161
 - of a subtree in a Fibonacci heap, 524
 - of a vertex cover, 1089, 1108
- skip list, 338
- slack, 855
- slack form, 846, 854–857
 - uniqueness of, 876
- slackness
 - complementary, 894 pr.
 - parallel, 781
- slack variable, 855
- slot
 - of a direct-access table, 254
 - of a hash table, 256
- SLOW-ALL-PAIRS-SHORTEST-PATHS, 689
- smoothed analysis, 897
- ★Socrates, 790
- solution
 - to an abstract problem, 1054
 - basic, 866
 - to a computational problem, 6
 - to a concrete problem, 1055
 - feasible, 665, 846, 851
 - infeasible, 851
 - optimal, 851
 - to a system of linear equations, 814
- sorted linked list, 236
 - see also* linked list
- sorting, 5, 16–20, 30–37, 147–212, 797–805
 - bubblesort, 40 pr.
 - bucket sort, 200–204
 - columnsort, 208 pr.
 - comparison sort, 191
 - counting sort, 194–197
 - fuzzy, 189 pr.
 - heapsort, 151–169
 - insertion sort, 12, 16–20

- k*-sorting, 207 pr.
 - lexicographic, 304 pr.
 - in linear time, 194–204, 206 pr.
 - lower bounds for, 191–194, 211, 531
 - merge sort, 12, 30–37, 797–805
 - by oblivious compare-exchange algorithms, 208 pr.
 - in place, 17, 148, 206 pr.
 - of points by polar angle, 1020 ex.
 - probabilistic lower bound for, 205 pr.
 - quicksort, 170–190
 - radix sort, 197–200
 - selection sort, 29 ex.
 - Shell's sort, 42
 - stable, 196
 - table of running times, 149
 - topological, 8, 612–615, 623
 - using a binary search tree, 299 ex.
 - with variable-length items, 206 pr.
 - 0-1 sorting lemma, 208 pr.
- sorting network, 811
- source vertex, 594, 644, 709, 712
- span law, 780
- spanning tree, 439, 624
 - bottleneck, 640 pr.
 - maximum, 1137 pr.
 - verification of, 642
 - see also* minimum spanning tree
- span, of a multithreaded computation, 779
- sparse graph, 589
 - all-pairs shortest paths for, 700–705
 - and Prim's algorithm, 638 pr.
- sparse-hulled distribution, 1046 pr.
- spawn**, in pseudocode, 776–777
- spawn edge, 778
- speedup, 780
 - of a randomized multithreaded algorithm, 811 pr.
- spindle, 485
- spine
 - of a string-matching automaton, 997 fig.
 - of a treap, 333 pr.
- splay tree, 338, 482
- spline, 840 pr.
- splitting
 - of B-tree nodes, 493–495
 - of 2-3-4 trees, 503 pr.
- splitting summations, 1152–1154
- spurious hit, 991
- square matrix, 1218
- SQUARE-MATRIX-MULTIPLY, 75, 689
- SQUARE-MATRIX-MULTIPLY-RECURSIVE, 77
- square of a directed graph, 593 ex.
- square root, modulo a prime, 982 pr.
- squaring, repeated
 - for all-pairs shortest paths, 689–691
 - for raising a number to a power, 956
- stability
 - numerical, 813, 815, 842
 - of sorting algorithms, 196, 200 ex.
- stack, 232–233
 - in Graham's scan, 1030
 - implemented by queues, 236 ex.
 - linked-list implementation of, 240 ex.
 - operations analyzed by accounting method, 457–458
 - operations analyzed by aggregate analysis, 452–454
 - operations analyzed by potential method, 460–461
 - for procedure execution, 188 pr.
 - on secondary storage, 502 pr.
- STACK-EMPTY, 233
- standard deviation, 1200
- standard encoding ($\{ \}$), 1057
- standard form, 846, 850–854
- star-shaped polygon, 1038 ex.
- start state, 995
- start time, 415
- state of a finite automaton, 995
- static graph, 562 n.
- static set of keys, 277
- static threading, 773
- stencil, 809 pr.
- stencil calculation, 809 pr.
- Stirling's approximation, 57
- storage management, 151, 243–244, 245 ex., 261 ex.
- store instruction, 23
- straddle, 1017
- strand, 777
 - final, 779
 - independent, 789
 - initial, 779
 - logically in parallel, 778

- logically in series, 778
- Strassen's algorithm, 79–83, 111–112
 - multithreaded, 795–796
- streaks, 135–139
- strictly decreasing, 53
- strictly increasing, 53
- string, 985, 1184
- string matching, 985–1013
 - based on repetition factors, 1012 pr.
 - by finite automata, 995–1002
 - with gap characters, 989 ex., 1002 ex.
 - Knuth-Morris-Pratt algorithm for, 1002–1013
 - naive algorithm for, 988–990
 - Rabin-Karp algorithm for, 990–995, 1013
- string-matching automaton, 996–1002, 1002 ex.
- strongly connected component, 1171
 - decomposition into, 615–621, 623
- STRONGLY-CONNECTED-COMPONENTS, 617
- strongly connected graph, 1171
- subgraph, 1171
 - predecessor, *see* predecessor subgraph
- subgraph-isomorphism problem, 1100 ex.
- subgroup, 943–946
- subpath, 1170
- subproblem graph, 367–368
- subroutine
 - calling, 21, 23, 25 n.
 - executing, 25 n.
- subsequence, 391
- subset (\subseteq), 1159, 1161
 - hereditary family of, 437
 - independent family of, 437
- SUBSET-SUM, 1097
- subset-sum problem
 - approximation algorithm for, 1128–1134, 1139
 - NP-completeness of, 1097–1100
 - with unary target, 1101 ex.
- substitution method, 83–88
 - and recursion trees, 91–92
- substring, 1184
- subtract instruction, 23
- subtraction of matrices, 1221
- subtree, 1176
 - maintaining sizes of, in order-statistic trees, 343–344
- success, in a Bernoulli trial, 1201
- successor
 - in binary search trees, 291–292
 - in a bit vector with a superimposed binary tree, 533
 - in a bit vector with a superimposed tree of constant height, 535
 - finding i th, of a node in an order-statistic tree, 344 ex.
 - in linked lists, 236
 - in order-statistic trees, 347 ex.
 - in proto van Emde Boas structures, 543–544
 - in red-black trees, 311
 - in Van Emde Boas trees, 550–551
- SUCCESSOR, 230
- such that ($:$), 1159
- suffix (\sqsupset), 986
- suffix function, 996
- suffix-function inequality, 999
- suffix-function recursion lemma, 1000
- sum (\sum), 1145
 - Cartesian, 906 ex.
 - infinite, 1145
 - of matrices, 1220
 - of polynomials, 898
 - rule of, 1183
 - telescoping, 1148
- SUM-ARRAYS, 805 pr.
- SUM-ARRAYS', 805 pr.
- summary
 - in a bit vector with a superimposed tree of constant height, 534
 - in proto van Emde Boas structures, 540
 - in van Emde Boas trees, 546
- summation, 1145–1157
 - in asymptotic notation, 49–50, 1146
 - bounding, 1149–1156
 - formulas and properties of, 1145–1149
 - linearity of, 1146
- summation lemma, 908
- supercomputer, 772
- superpolynomial time, 1048
- supersink, 712
- supersource, 712
- surjection, 1167
- SVD, 842
- sweeping, 1021–1029, 1045 pr.
 - rotational, 1030–1038

- sweep line, 1022
- sweep-line status, 1023–1024
- symbol table, 253, 262, 265
- symmetric difference, 763 pr.
- symmetric matrix, 1220, 1222 ex., 1226 ex.
- symmetric positive-definite matrix, 832–835, 842
- symmetric relation, 1163
- symmetry of Θ -notation, 52
- sync**, in pseudocode, 776–777
- system of difference constraints, 664–670
- system of linear equations, 806 pr., 813–827, 840 pr.

- TABLE-DELETE, 468
- TABLE-INSERT, 464
- tail
 - of a binomial distribution, 1208–1215
 - of a linked list, 236
 - of a queue, 234
- tail recursion, 188 pr., 419
- TAIL-RECURSIVE-QUICKSORT, 188 pr.
- target, 1097
- Tarjan's off-line least-common-ancestors algorithm, 584 pr.
- task, 443
- Task Parallel Library, 774
- task scheduling, 443–446, 448 pr., 450
- tautology, 1066 ex., 1086 ex.
- Taylor series, 306 pr.
- telescoping series, 1148
- telescoping sum, 1148
- testing
 - of primality, 965–975, 983
 - of pseudoprimality, 966–968
- text, in string matching, 985
- then** clause, 20 n.
- Theta-notation, 44–47, 64
- thread, 773
- Threading Building Blocks, 774
- 3-CNF, 1082
- 3-CNF-SAT, 1082
- 3-CNF satisfiability, 1082–1085, 1105
 - approximation algorithm for, 1123–1124, 1139
 - and 2-CNF satisfiability, 1049
- 3-COLOR, 1103 pr.
- 3-conjunctive normal form, 1082
- tight constraint, 865
- time, *see* running time
- time domain, 898
- time-memory trade-off, 365
- timestamp, 603, 611 ex.
- Toeplitz matrix, 921 pr.
- to**, in pseudocode, 20
- TOP, 1031
- top-down method, for dynamic programming, 365
- top of a stack, 232
- topological sort, 8, 612–615, 623
 - in computing single-source shortest paths in a dag, 655
- TOPOLOGICAL-SORT, 613
- total order, 1165
- total path length, 304 pr.
- total preorder, 1165
- total relation, 1165
- tour
 - bitonic, 405 pr.
 - Euler, 623 pr., 1048
 - of a graph, 1096
- track, 486
- tractability, 1048
- trailing pointer, 295
- transition function, 995, 1001–1002, 1012 ex.
- transitive closure, 697–699
 - and boolean matrix multiplication, 832 ex.
 - of dynamic graphs, 705 pr., 707
- TRANSITIVE-CLOSURE, 698
- transitive relation, 1163
- transitivity of asymptotic notation, 51
- TRANSPLANT, 296, 323
- transpose
 - conjugate, 832 ex.
 - of a directed graph, 592 ex.
 - of a matrix, 1217
 - of a matrix, multithreaded, 792 ex.
- transpose symmetry of asymptotic notation, 52
- traveling-salesman problem
 - approximation algorithm for, 1111–1117, 1139
 - bitonic euclidean, 405 pr.
 - bottleneck, 1117 ex.
 - NP-completeness of, 1096–1097
 - with the triangle inequality, 1112–1115
 - without the triangle inequality, 1115–1116

- traversal of a tree, 287, 293 ex., 342, 1114
- treap, 333 pr., 338
- TREAP-INSERT, 333 pr.
- tree, 1173–1180
 - AA-trees, 338
 - AVL, 333 pr., 337
 - binary, *see* binary tree
 - binomial, 527 pr.
 - bisection of, 1181 pr.
 - breadth-first, 594, 600
 - B-trees, 484–504
 - decision, 192–193
 - depth-first, 603
 - diameter of, 602 ex.
 - dynamic, 482
 - free, 1172–1176
 - full walk of, 1114
 - fusion, 212, 483
 - heap, 151–169
 - height-balanced, 333 pr.
 - height of, 1177
 - interval, 348–354
 - k -neighbor, 338
 - minimum spanning, *see* minimum spanning tree
 - optimal binary search, 397–404, 413
 - order-statistic, 339–345
 - parse, 1082
 - recursion, 37, 88–93
 - red-black, *see* red-black tree
 - rooted, 246–249, 1176
 - scapegoat, 338
 - search, *see* search tree
 - shortest-paths, 647–648, 673–676
 - spanning, *see* minimum spanning tree, spanning tree
 - splay, 338, 482
 - treap, 333 pr., 338
 - 2-3, 337, 504
 - 2-3-4, 489, 503 pr.
 - van Emde Boas, 531–560
 - walk of, 287, 293 ex., 342, 1114
 - weight-balanced trees, 338
- TREE-DELETE, 298, 299 ex., 323–324
- tree edge, 601, 603, 609
- TREE-INSERT, 294, 315
- TREE-MAXIMUM, 291
- TREE-MINIMUM, 291
- TREE-PREDECESSOR, 292
- TREE-SEARCH, 290
- TREE-SUCCESSOR, 292
- tree walk, 287, 293 ex., 342, 1114
- trial, Bernoulli, 1201
- trial division, 966
- triangle inequality, 1112
 - for shortest paths, 650, 671
- triangular matrix, 1219, 1222 ex., 1225 ex.
- trichotomy, interval, 348
- trichotomy property of real numbers, 52
- tridiagonal linear systems, 840 pr.
- tridiagonal matrix, 1219
- trie (radix tree), 304 pr.
 - y-fast, 558 pr.
- TRIM, 1130
- trimming a list, 1130
- trivial divisor, 928
- truth assignment, 1072, 1079
- truth table, 1070
- TSP, 1096
- tuple, 1162
- twiddle factor, 912
- 2-CNF-SAT, 1086 ex.
- 2-CNF satisfiability, 1086 ex.
 - and 3-CNF satisfiability, 1049
- two-pass method, 571
- 2-3-4 heap, 529 pr.
- 2-3-4 tree, 489
 - joining, 503 pr.
 - splitting, 503 pr.
- 2-3 tree, 337, 504
- unary, 1056
- unbounded linear program, 851
- unconditional branch instruction, 23
- uncountable set, 1161
- underdetermined system of linear equations, 814
- underflow
 - of a queue, 234
 - of a stack, 233
- undirected graph, 1168
 - articulation point of, 621 pr.
 - biconnected component of, 621 pr.
 - bridge of, 621 pr.
 - clique in, 1086
 - coloring of, 1103 pr., 1180 pr.

- computing a minimum spanning tree in, 624–642
 - converting to, from a multigraph, 593 ex.
 - d -regular, 736 ex.
 - grid, 760 pr.
 - hamiltonian, 1061
 - independent set of, 1101 pr.
 - matching of, 732
 - nonhamiltonian, 1061
 - vertex cover of, 1089, 1108
 - see also* graph
- undirected version of a directed graph, 1172
- uniform hashing, 271
- uniform probability distribution, 1191–1192
- uniform random permutation, 116, 125
- union
 - of dynamic sets, *see* uniting
 - of languages, 1058
 - of sets (\cup), 1159
- UNION, 505, 562
 - disjoint-set-forest implementation of, 571
 - linked-list implementation of, 565–567, 568 ex.
- union by rank, 569
- unique factorization of integers, 931
- unit (1), 928
- uniting
 - of Fibonacci heaps, 511–512
 - of heaps, 506
 - of linked lists, 241 ex.
 - of 2-3-4 heaps, 529 pr.
- unit lower-triangular matrix, 1219
- unit-time task, 443
- unit upper-triangular matrix, 1219
- unit vector, 1218
- universal collection of hash functions, 265
- universal hashing, 265–268
- universal sink, 593 ex.
- universe, 1160
 - of keys in van Emde Boas trees, 532
- universe size, 532
- unmatched vertex, 732
- unsorted linked list
 - see also* linked list
- until**, in pseudocode, 20
- unweighted longest simple paths, 382
- unweighted shortest paths, 381
- upper bound, 47
- upper-bound property, 650, 671–672
- upper median, 213
- upper square root ($\sqrt{}$), 546
- upper-triangular matrix, 1219, 1225 ex.
- valid shift, 985
- value
 - of a flow, 710
 - of a function, 1166
 - objective, 847, 851
- value over replacement player, 411 pr.
- Vandermonde matrix, 902, 1226 pr.
- van Emde Boas tree, 531–560
 - cluster in, 546
 - compared with proto van Emde Boas structures, 547
 - deletion from, 554–556
 - insertion into, 552–554
 - maximum in, 550
 - membership in, 550
 - minimum in, 550
 - predecessor in, 551–552
 - with reduced space, 557 pr.
 - successor in, 550–551
 - summary in, 546
- Var [] (variance), 1199
- variable
 - basic, 855
 - entering, 867
 - leaving, 867
 - nonbasic, 855
 - in pseudocode, 21
 - random, 1196–1201
 - slack, 855
 - see also* indicator random variable
- variable-length code, 429
- variance, 1199
 - of a binomial distribution, 1205
 - of a geometric distribution, 1203
- VEB-EMPTY-TREE-INSERT, 553
- vEB tree, *see* van Emde Boas tree
- VEB-TREE-DELETE, 554
- VEB-TREE-INSERT, 553
- VEB-TREE-MAXIMUM, 550
- VEB-TREE-MEMBER, 550
- VEB-TREE-MINIMUM, 550
- VEB-TREE-PREDECESSOR, 552
- VEB-TREE-SUCCESSOR, 551

- vector, 1218, 1222–1224
 - convolution of, 901
 - cross product of, 1016
 - orthonormal, 842
 - in the plane, 1015
- Venn diagram, 1160
- verification, 1061–1066
 - of spanning trees, 642
- verification algorithm, 1063
- vertex
 - articulation point, 621 pr.
 - attributes of, 592
 - capacity of, 714 ex.
 - in a graph, 1168
 - intermediate, 693
 - isolated, 1169
 - overflowing, 736
 - of a polygon, 1020 ex.
 - reabeled, 740
 - selector, 1093
- vertex cover, 1089, 1108, 1124–1127, 1139
- VERTEX-COVER, 1090
- vertex-cover problem
 - approximation algorithm for, 1108–1111, 1139
 - NP-completeness of, 1089–1091, 1105
- vertex set, 1168
- violation, of an equality constraint, 865
- virtual memory, 24
- Viterbi algorithm, 408 pr.
- VORP, 411 pr.
- walk of a tree, 287, 293 ex., 342, 1114
- weak duality, 880–881, 886 ex., 895 pr.
- weight
 - of a cut, 1127 ex.
 - of an edge, 591
 - mean, 680 pr.
 - of a path, 643
- weight-balanced tree, 338, 473 pr.
- weighted bipartite matching, 530
- weighted matroid, 439–442
- weighted median, 225 pr.
- weighted set-covering problem, 1135 pr.
- weighted-union heuristic, 566
- weighted vertex cover, 1124–1127, 1139
- weight function
 - for a graph, 591
 - in a weighted matroid, 439
- while**, in pseudocode, 20
- white-path theorem, 608
- white vertex, 594, 603
- widget, 1092
- wire, 1071
- WITNESS, 969
- witness, to the compositeness of a number, 968
- work law, 780
- work, of a multithreaded computation, 779
- work-stealing scheduling algorithm, 812
- worst-case running time, 27, 49
- Yen's improvement to the Bellman-Ford algorithm, 678 pr.
- y-fast trie, 558 pr.
- Young tableau, 167 pr.
- \mathbb{Z} (set of integers), 1158
- \mathbb{Z}_n (equivalence classes modulo n), 928
- \mathbb{Z}_n^* (elements of multiplicative group modulo n), 941
- \mathbb{Z}_n^+ (nonzero elements of \mathbb{Z}_n), 967
- zero matrix, 1218
- zero of a polynomial modulo a prime, 950 ex.
- 0-1 integer programming, 1100 ex., 1125
- 0-1 knapsack problem, 425, 427 ex., 1137 pr., 1139
- 0-1 sorting lemma, 208 pr.
- zonk, 1195 ex.