

Index

- Abstract transitive closure, 174-175, 217-221
- Active vertex, 411
- Acyclic graph. *See* Digraph; Directed acyclic graph (DAG)
- Acyclic network, 313-321, 334-335
 - maxflow, 427-429
- Adjacency-lists representation, 31-35
 - DFS, 90-91, 95, 101
 - digraphs, 37, 153, 155, 179
 - find/remove edge, 41-42
 - flow networks, 379
 - performance, 35, 37-40, 142-145
 - standard adjacency-lists DFS, 96, 161
 - transitive closure, 174
 - weighted graphs/networks, 37, 230, 235-238, 278
- Adjacency-matrix representation, 25-30
 - DFS, 93
 - digraphs, 37, 153-155, 169-172, 176, 179
 - flow networks, 379
 - linear-time algorithm, 59
 - performance, 29-30, 37-40, 142-144
 - removing vertices, 41-42
 - standard adjacency-matrix DFS, 96-97, 161
 - weighted graphs/networks, 37, 230, 233-238, 278
- Adjacent vertices, 9
- Adjlist iterator, 17-18, 29
- ADT, graph. *See* Graph ADT
- ADT interfaces
 - all-pairs shortest paths, 305
 - connectivity, 22
 - graph, 17
 - graph-processing input/output, 21
 - weighted graphs, 231
- Airline route, 283
- All-pairs shortest path, 281, 304-311
 - acyclic networks, 318-320
 - BFS, 127-128
 - negative weights, 356-360
 - path relaxation, 288-290
 - and transitive closure, 175, 328-329
- and* operation, 169-170
- Arbitrage, 348-349
- Arbitrary weight, 228
- Arc, 8, 14
- Array
 - of arrays, 29
 - of bits, 28
 - of edges, 24, 37-38, 40
 - isomorphic graphs, 29
 - of lists, 31-32
 - vertex-indexed, 37-38
- Articulation point, 117-119
- Assignment problem, 74, 476-477
- Augmenting-path method, 382-407
 - cycle canceling, 451
 - longest augmenting path, 396
 - maximum-capacity, 394, 399-401, 405-406
 - network simplex algorithm, 460-461
 - performance, 393, 395-406, 434-436
 - and preflow-push, 411
 - random flow networks, 402-404
 - randomized, 402
 - shortest augmenting path, 393-394, 398-399, 405-407
 - stack-based, 400
- A* algorithm, 325
- Back edge, 99, 113-114, 161-165, 202-203
- Back link, 99-100
- Bellman, R., 350
- Bellman-Ford algorithm, 350-356, 359-360, 447
- BFS tree, 123-129
- Binary DAG, 189-190
- Binary decision diagram (BDD), 189
- Binary tree, 189-190
- Bioconnectivity, 117-119
- Bipartite graph, 13-14, 109-110
- Bipartite matching, 74, 433-436, 476, 482
- Boolean matrix multiplication, 169-172, 176-177
- Boruvka's algorithm, 244, 264-267
- Breadth-first search (BFS), 81-82, 121-130
 - and DFS, 127-134
 - forest, 125
 - fringe size, 137
 - PFS, 255, 302
 - tree, 123-129
- Bridge, 112-116
- Bridges of Königsberg problem, 62-64
- Capacity, flow, 372, 375
 - st*-cut, 385
 - vertex-capacity constraints, 426
- change priority* operation, 255
- Circuit, 4
- Circulation, flow network, 377-378, 429
- Class
 - augmenting-paths, 391
 - cycle canceling, 449
 - DFS, 95-96

- flow networks, 379, 432
 - network simplex algorithm, 466-467
 - weighted graphs, 233-236
- Clique, 13, 76
- Clone implementation, 30, 33-34
- Colorability problem, 76
 - two-coloring, 109-110
- Communications network, 367-369
- Complement, 12
- Complete graph, 12
- Connection, 3
- Connectivity, 11-12
 - bioconnectivity, 117-119
 - digraphs, 158
 - edge, 437-439
 - equivalence relations, 184
 - general connectivity, 74-75, 119
 - interface, 22
 - k -connected graph, 118-119
 - maxflow, 437-438
 - random, 141-146
 - simple connectivity, 72, 106-108
 - st -connectivity, 119
 - strong connectivity, 72, 156-158, 205-206
 - vertex connectivity, 118-119, 438
- Constraint, 186
- Cost, 227, 372, 443
 - convex/nonlinear, 483
 - edge, 474
 - flow, 443
 - negative, 446-447
 - reduced, 454-457
 - see also* Performance; Weighted graph
- Critical edge, 398-399
- Cross edge, 161-162, 202-203
- Crossing edge, 241, 385
- Cut, 241
 - mincut, 386-387
 - property, 241, 245
 - set, 385
 - st -cut, 385-387, 427-428
 - vertex, 117-118
- Cut problem, 369-370, 385-386
- Cycle, 10-11
 - DFS, 105
 - directed, 14-15, 155, 162-165
 - even, 74, 224-225
 - flow networks, 377-378, 427
 - MSTs, 241-244
 - negative, 279-280, 347-350, 446-454
 - odd, 109-110
 - property, 242-243, 245
- Cycle-canceling algorithm, 372, 447-452
 - see also* Network simplex algorithm
- Cyclic path, 10, 62-64, 156, 477-478
- DAG. *See* Directed acyclic graph (DAG)
- de Bruijn graph, 53
- decrease key* operation, 256, 268-269, 297
- Degrees-of-separation graph, 53
- Delauney triangulation, 274-275
- Dense graph, 13, 25, 29-30, 251
- Depth-first search (DFS), 81-82, 87-91
 - and BFS, 127-134
 - classes, 95-96
 - cycle detection, 105
 - digraphs, 160-168
 - fringe size, 137
 - PFS, 255
 - running time, 93-95
 - simple connectivity, 106-108
 - simple path, 57-60, 106
 - spanning forest, 92-94, 109
 - standard adjacency representations, 96-97, 161
 - strong components algorithms, 207-215
 - topological sorting, 193-197, 202-203
 - transitive closure, 177-179
 - tree links, 99-100
- and Trémaux exploration, 87-90
- two-coloring, 109-110
- two-way Euler tour, 68, 108-109
- and union-find, 107-108
- vertex search, 109
 - see also* DFS tree
- Destination vertex, 14
- DFS forest, 92-94, 97-103, 123-125
 - DAGs, 194
 - digraphs, 161-162, 165
 - spanning forest, 109
- DFS tree, 90, 97-103
 - bridges, 113-116
 - diagraphs, 161-162
- d -heap, 271-272
- Diameter, network, 304
- Difference constraints, 333-336, 338, 347
- Digraph, 14-15, 149-225
 - adjacency-lists representation, 36-37, 153, 155, 179
 - adjacency-matrix representation, 36-37, 153-155, 169-172, 176, 179
 - connectivity, 158, 438
 - decomposing, 166
 - defined, 152
 - DFS in, 160-168
 - directed cycle detection, 162-166, 187-188
 - directed path, 152, 155
 - edge direction, 149-150
 - even cycles, 74
 - grid, 152
 - isomorphism, 150
 - map, 154
 - random, 220
 - and relations, 182-183
 - reverse, 154-155
 - running time, 222-223
 - single-source reachability, 166-167
 - strong components, 157-158, 205-215

- strongly connected, 72, 156-158, 205-206
- transitive closure, 169-180, 217-221
- and undirected graphs, 153, 155-162, 164-167, 179-180
- unconnected, 224
- weighted, 277-278
- Dijkstra's algorithm, 257, 293-302
 - acyclic networks, 315, 320
 - all-pairs shortest paths, 305-311
 - Euclidean networks, 323-326
 - negative weights, 349, 354-360
 - and Prim's algorithm, 294-296
- Directed acyclic graph (DAG), 15, 150, 186-204
 - binary, 189-190
 - defined, 155
 - detection, 162-165, 187-188
 - dominators, 224
 - kernel, 157, 217-221
 - longest path, 199
 - partial orders, 184-185
 - scheduling problem, 187
 - sink/source, 197-199
 - strong components, 157
 - topological sorting, 187, 191-200
 - transitive closure, 202-204
 - weighted, 313-320
- Directed cycle, 14-15, 155, 162-165, 187-188
- Directed graph. *See* Digraph; Directed acyclic graph (DAG)
- Directed path, 72, 152, 155, 224
- Disjoint paths, 11, 117
- Distances matrix, 289, 306, 360
- Distribution network, 444-445
- Distribution problem, 368, 430, 444-445, 475
- Dominator, 224
- Down link, 99-100, 161-162, 202-203
- Dummy edge, 447-451, 454-455, 472
- Dummy vertex, 376-377
- Dynamic algorithm, 22-23, 50
- Dynamic reachability, 222-223
- Edge, 7-10
 - array of edges, 24
 - back, 99, 113-114, 161-165, 202-203
 - backward/forward, 383-384
 - connectivity, 437-439
 - costs, 474
 - critical, 398-399
 - cross, 161-162, 202-203
 - crossing, 241, 385
 - directed, 14
 - disjoint, 11, 117, 437
 - down, 99-100, 161-162, 202-203
 - dummy, 447-451, 454-455, 472
 - eligible, 412, 456-457, 462-468
 - flows, 375
 - incident, 9
 - marking vertices, 92
 - parallel, 8, 19-20, 28, 40-41, 47-48, 237, 278
 - random, 47-48
 - references, 231-232, 235-238, 277-278, 379, 432
 - relaxation, 286-287, 292, 323
 - separation, 113
 - tree, 99, 113-114, 161-162, 202
 - un capacitated, 380
 - weights, 234-235, 482
- Edge data type, 17, 37, 231-232, 379, 388-390
- Edge-based preflow-push algorithm, 413-414
- Edge-connected graph, 112-113, 117-119, 438-439
- Edge-separable graph, 113
- Edmonds, J., 393
- Eligible edge, 412, 456-457, 462-468
- Encapsulation, 39
- Enumeration, graph, 150-151
- Equal weights, 228-229
- Equivalence class, 183-184
- Equivalence relation, 183
- Equivalent problems, 328
- Erdős, P., 142
- Euclidean graph, 10
 - flow networks, 402-404
 - MSTs, 274-276
 - neighbor graphs, 49
 - networks, 322-326
- Euclidean heuristic, 324-326
- Euler path, 62-68
 - directed, 224
 - mail carrier, 477-478
 - two-way Euler tour, 108-109
- Even cycle, 74, 224-225
- Excess, vertex, 411
- Existence problem, 76
- Fat interface, 39
- Feasible flow, 410, 430-432, 444
- Feasible spanning tree, 453-455
- Feedback vertex set, 224
- Fibonacci computation, 188, 268-269
- FIFO queue, 121-125, 132-138
 - PFS, 302
 - preflow-push, 416-420
 - topological sort, 199-200
- find edge* operation, 40-41
- Flow cost, 443
- Flow network, 15, 367-486
 - augmenting-flow method, 382-407
 - backward/forward edges, 383-384
 - capacities, 372, 375, 385, 426
 - circulation, 377-378, 429
 - cut problems, 369-370, 385-386
 - defined, 375
 - distribution problems, 368, 430, 444-445, 475
 - equilibrium, 376-377
 - flows, 372, 375
 - inflow/outflow, 375-377
 - matching problems, 369, 433-436, 476, 482
 - max-flow-min-cut theorem, 385-387

- model, 373-374
- network simplex algorithm, 372, 453-470, 479
- preflow-push method, 410-423
- random, 402-404
- reductions, 367, 406, 425-440
- representations, 379
- residual, 388-391, 412, 417, 446, 453
- st*-network, 375-377, 429-430
- value, 375-377
- see also* Maxflow problem; Mincost flow problem; Network
- Floyd's algorithm, 176, 289-290, 304, 307-311
- negative weights, 349-351, 354
- Ford, L. R., 350, 382
- Ford-Fulkerson method. *See* Augmenting-path method
- Forest, 12
- BFS, 125
- Boruvka's algorithm, 264-265
- DFS, 92-94, 97-103, 109, 123-125, 161-162, 165, 194
- Kruskal's algorithm, 259
- spanning, 109
- see also* DFS tree; Tree
- Four-color theorem, 76
- Fringe, 131-138, 296
- Prim's algorithm, 251-257
- priority queue, 393
- Fulkerson, D. R., 382

- Gabow's algorithm, 207, 212-215
- General connectivity, 74-75, 119
- Generalized queue, 416, 418
- Geometric algorithm, 276
- Goldberg, A., 410
- Graph, 3-80
- applications, 4-5
- bipartite, 13, 109-110
- complete, 12
- connected, 11-12
- de Bruijn, 53
- defined, 7
- degrees-of-separation, 53
- dense/sparse, 13, 25, 29-30
- directed/undirected, 14-15
- edge-connected/edge-separable, 112-113, 117
- Euclidean, 10
- interval, 53
- isomorphic, 10, 29
- method invocation, 50-53
- multigraph, 8
- neighbor, 49
- planar, 9, 73
- random, 48-49
- simple, 8
- static, 42, 50
- subgraph, 9
- transaction, 50
- weighted, 15
- see also* Digraph; Directed acyclic graph (DAG); Particular graph types
- Graph ADT, 16-25
- adjacency-lists representation, 31-35
- adjacency-matrix representation, 25-30
- all-pairs shortest paths, 304-305
- equivalence relations, 184
- graph-search functions, 91-97
- symbol table, 51
- weighted graphs, 230-239
- Graph search, 81-146
- ADT functions, 91-97
- algorithm analysis, 140-146
- bioconnectivity, 117-119
- generalized, 131-138
- Hamilton tour, 60
- maze exploration, 82-86
- priority-first search, 252-256, 296-302, 322-324, 392-393
- randomized, 136-138
- separability, 112-116
- simple path, 57-60, 61, 106
- see also* Breadth-first search (BFS); Depth-first search (DFS)
- Graph-processing problem, 20-25, 71
- degree of difficulty, 71, 77-79
- existence, 76
- intractable, 75
- NP-hard, 75, 77, 339-340
- tractable, 73
- Grid digraph, 152

- Hamilton path, 60-62, 224, 339-340
- Height function, 412-415
- Highest-vertex preflow-push, 420
- Hypertext, 4

- Immediate dominator, 224
- Incident edge, 9
- Indegree, 14, 153-154, 197
- Independent set problem, 76
- Indexing function, 52
- Induced subgraph, 9
- Infeasible problem, 336
- Inflow, 375-377
- Inheritance, 39
- Initial height function, 413
- Integer weights, 379-380
- Interface, fat, 39
- see also* ADT interfaces
- Intersection, 82
- Interval graph, 53
- Intractable problem, 75, 342
- Irreflexive relation, 183
- Isomorphic graphs, 10, 29, 79, 150
- Item, 3

- Jarnik's algorithm, 257
- Job placement, 369, 476
- Job scheduling, 4, 150, 187
- negative cycles, 347-350
- shortest paths, 332-338, 344
- see also* Topological sorting
- Johnson, D., 269

- Johnson's algorithm, 360
- Karp, R., 207, 393
- k -connected graph, 118-119
- Kernel DAG, 157, 217-221
- k -neighbor graph, 49
- Königsberg, bridges of, 62-64
- Kosaraju's algorithm, 207-210
- Kruskal's algorithm, 243-244, 258-264, 268
- Kuratowski's theorem, 73
- Least common ancestor (LCA), 459-460
- Length, path, 11
- < operator, 290
- Library, software, 485
- Library programming, 336
- LIFO stack, 132-134
- Linear programming (LP), 332-333, 342-343, 365
- maxflow, 439-440
- mincost flow, 425, 479
- multicommodity flow, 483
- network flow algorithms, 371-372
- Linear quantity, 59
- Link, 4, 8
- back, 99
- DFS tree, 99-100
- down, 99-100, 161-162, 202-203
- parent, 98-100, 164
- Locality property, 49
- Longest paths, 75-76, 199, 315-320
- augmenting paths, 396
- difference constraints, 334-335
- and shortest path, 329-330
- Mail carrier, 477-478
- Map, 4, 154, 326
- Matching, 5, 73-74
- bipartite, 74, 433-436, 476
- maximum, 482
- maximum-cardinality, 433, 482
- minimum-distance point matching, 369
- Mathematical programming, 343
- Matrix, adjacency, 25-30
- Maxflow problem, 372, 375-378, 382-440
- acyclic networks, 427-429
- augmenting-path method, 382-407
- bipartite matching, 433-436
- capacity constraints, 426
- connectivity, 437-438
- feasible flow, 430-432
- general networks, 425-426
- linear programming, 439-440
- and mincost flow problem, 425, 472-473
- mincost maxflow, 443-447
- preflow-push method, 410-423
- reductions, 370, 406, 425-440
- running time, 434-436
- spanning tree, 454-455
- standard, 425-426
- undirected networks, 429-430
- Maxflow-mincut theorem, 385-387, 437
- Maximal connected subgraph, 12
- Maximum-capacity augmenting path, 394, 399-401, 405-406
- Maximum-cardinality matching, 433, 482
- Maze, 82-86
- Menger's theorem, 119, 437
- Merchandise distribution, 368, 430, 444-445, 475
- Method invocation graph, 50-53
- Mincost flow problem, 372, 443-479
- assignment problem, 476-477
- cycle-canceling algorithm, 372, 447-452
- edge costs, 474
- eligible edges, 456-457
- feasible flow, 444-445, 472-473
- flow cost, 443
- LP formulation, 425, 479
- mail carrier, 477-478
- and maxflow problem, 425, 472-473
- mincost maxflow, 443-447
- reductions, 472-479
- running time, 451
- single-source shortest paths, 472-473
- transportation, 475-476
- see also* Network simplex algorithm
- Mincut, 386-387
- Minimum spanning tree (MST), 72-73, 227-276
- Boruvka's algorithm, 244, 264-267
- cut and cycle properties, 241-244
- defined, 228
- equal weights, 228-229
- Euclidean, 274-276
- Kruskal's algorithm, 243-244, 258-264, 268
- performance, 268-273, 269-271
- Prim's algorithm, 243, 247-257, 263, 269
- representations, 237-239
- weighted-graph ADTs, 230-239
- Minimum-distance point matching, 369
- Modular arithmetic, 184
- Multicommodity flow, 483
- Multigraph, 8
- Multisource paths, 314-318
- Negative cost, 446-447
- Negative cycle, 279-280, 347-350, 446-454
- Negative weight, 345-365
- arbitrage, 348-349
- Bellman-Ford algorithm, 350-356, 358
- Dijkstra's algorithm, 349, 354-360

- Floyd's algorithm, 349-351, 354
- Neighbor graph, 49, 142
- Network, 5, 15, 277-284
 - acyclic, 313-320, 334-335, 427-429
 - adjacency representations, 37
 - communications, 368-369
 - distribution, 444-445
 - reliability, 370
 - residual, 388-391, 412, 417, 446, 453
 - reweighting, 324, 356-360
 - st*-network, 375-378, 429-430
 - telephone, 370
 - undirected, 429-430
 - weighted diameter, 304
 - see also* Flow network; Shortest path
- Network simplex algorithm, 372, 453-470
 - assignment problem, 476-477
 - eligible edges, 456-457, 462-468
 - feasible spanning tree, 453-455
 - implementations, 466-469
 - initialization, 464
 - parent-link representation, 458-464
 - performance, 466-470
 - shortest paths, 472-473
 - simplex method, 479
 - vertex potentials, 454-456, 468
 - see also* Mincost flow problem
- Node, 8
- Nonlinear cost, 483
- NP-hard problem, 75, 77, 339-340

- Online algorithm, 22-23
- Operations research (OR), 343, 365
- Optimality condition, 242
- Optimization, 77
- or* operation, 169-170
- Ordered pair, 14, 183
- Outdegree, 14, 153-154

- Outflow, 375-378

- Parallel edges, 8, 19-20, 28, 40-41
 - networks, 278
 - random graphs, 47-48
 - weighted graphs, 237
- Parent-link representation
 - BFS tree, 127
 - cycle detection, 164
 - DFS tree, 98-100
 - network simplex algorithm, 458-464
- Partial order, 184-185
- Passage, 82
- Path, 10-11, 57-68
 - cyclic, 10, 156, 477-478
 - directed, 72, 152, 155, 224
 - disjoint, 11, 117
 - Euler, 62-68, 108-109
 - Hamilton, 60-62, 224, 339-340
 - mail carrier, 477-478
 - relaxation, 286, 288-290, 292
 - simple, 57-61, 106, 279
 - weight, 277
 - see also* Longest path; Shortest path
- Paths matrix, 289-290, 306, 360
- Performance, 6, 79
 - abstraction, 43
 - adjacency-lists representation, 35, 37-40, 142-145
 - adjacency-matrix representation, 29-30, 37-40, 142-144
 - array-of-edges, 37-38, 40
 - augmenting-path method, 393, 395-406, 434-436
 - cycle canceling, 452
 - dense/sparse graphs, 13
 - DFS forests, 102-103
 - Dijkstra's algorithm, 297-300, 324-326
 - equivalent problems, 330-331
 - Kruskal's algorithm, 259-263
 - MST algorithms, 268-273
 - network simplex algorithm, 466-470
 - path search, 61-62, 64
 - PFS, 256-257, 297-300
 - preflow-push, 419-423
 - preprocessing time, 106-107, 127, 217, 222
 - shortest-paths algorithms, 363-365
 - static graphs, 42
 - transitive closure, 179-180, 222
 - union-find algorithm, 146
 - worst-case, 43-44
 - see also* Running time
- PERT chart, 150
- Planar graph, 9, 73
- Planarity problem, 73
- + operator, 290
- Polynomial-time reduction, 439
- Postman problem, 75, 224
- Postorder numbering, 101, 162, 193-194
- Potential function, 416-419, 454-456, 468
- Precedence constraint, 187, 332
- Preflow, 410-411
- Preflow-push method, 410-423
 - edge-based, 413-414
 - highest-vertex, 420
 - performance, 419-423
 - vertex-based, 416
- Preorder numbering, 99, 101, 162
- Preprocessing, 22, 106-107, 127, 217, 222
 - shortest paths, 304-305
- Prim's algorithm, 243, 247-257, 263
 - and Dijkstra's algorithm, 294-296
 - running time, 269
- Priority queue, 137, 269-271
 - augmenting-path method, 392-393
 - Kruskal's algorithm, 262-263
 - multiway heap, 272
- Priority-first search (PFS), 322-324
 - augmenting-path method, 392-393
 - Dijkstra's algorithm, 296-302

- Prim's algorithm, 252-256
- Program structure, 5
- Pushdown stack, 121

- Query methods, 22, 39
- Quicksort, 261-262, 269

- Radar tracking system, 369
- Radix sort, 261, 269
- Random flow network, 402-404
- Random graph, 48-49, 141-146
- Randomized queue, 136-137, 402
- Reachability, 152, 158, 166-167
 - DAGs, 202-203
 - digraphs, 205
 - dynamic, 222-223
 - see also* Transitive closure
- Reduced cost, 454-457
- Reduction, 177, 283, 484
 - difference constraints, 333-336, 338
 - equivalent problems, 328
 - flow networks, 367
 - implications, 341
 - job scheduling, 332-338
 - linear programming, 332-333, 342-343
 - maxflow, 370, 406, 425-440
 - mincost flow, 472-479
 - polynomial-time, 439
 - shortest paths, 328-343
 - transitive, 224
 - upper/lower bounds, 342
- Reference, edge, 231-232, 235-238, 277-278, 379, 432
- Reflexive relation, 183
- Relation, 183
- Relaxation, 285-292, 323
- remove edge* operation, 34, 40-41
- remove the minimum* operation, 251, 268-271
- Renyi, A., 142
- Residual network, 388-391, 453
 - mincost flow, 446
 - preflow-push, 412, 417
- Reverse topological sorting, 193-197
- Reweighting, 324, 356-360
- Road map, 282-283
- Running time, 141
 - augmenting-path method, 393, 405
 - Bellman-Ford algorithm, 354-356
 - Boruvka's algorithm, 265-266
 - DFS, 93-95
 - digraphs, 222-223
 - Dijkstra's algorithm, 296, 299, 309-311
 - Floyd's algorithm, 309-311
 - Kruskal's algorithm, 259-263
 - maxflow, 434-436
 - mincost flow, 451
 - MST algorithms, 269-270
 - NP-hard problems, 339
 - path search, 61-62
 - preflow-push, 416-419
 - Prim's algorithm, 269
 - random graph connectivity, 142
 - shortest path algorithms, 301
 - strong components in digraphs, 206
 - transitive closure, 172-174, 177, 180, 222
 - weighted graphs, 234
 - see also* Performance

- Scheduling problem, 4, 150, 483
 - DAGs, 187
 - precedence-constrained, 332-338
 - see also* Topological sorting
- Search. *See* Graph search
- Selection, 331
- Self-loop, 8, 19, 28
 - digraphs, 152
 - networks, 278
 - relations, 183
- Sentinel weight, 234-235, 278, 287

- Separability, 112-116
- Separation vertex, 117-118
- Set-inclusion DAG, 184
- Shortest path, 176, 277-365
 - acyclic networks, 313-320
 - augmenting paths, 393-394, 398-399, 405-407
 - Bellman-Ford algorithm, 350-356, 358-360
 - BFS, 121
 - defined, 279
 - Dijkstra's algorithm, 293-302, 305-311, 349, 354-360
 - Euclidean networks, 322-326
 - Floyd's algorithm, 304, 307-311, 349-351, 354-356
 - and longest path, 329-330
 - multisource, 314-318
 - negative weights, 345-361
 - network simplex algorithm, 472-473
 - NP-hard problems, 339-340
 - performance, 301, 363-365
 - reduction, 328-343
 - relaxation, 285-292
 - shortest paths tree (SPT), 279, 287-288, 294
 - source-sink, 281, 294, 322-326
 - terminology, 286, 313
 - see also* All-pairs shortest path; Network; Single-source shortest path
- Simple connectivity, 72, 106-108
- Simple graph, 8
- Simple path, 10, 57-61
 - DFS, 106
 - networks, 279
- Simplex method, 479
- Single-source longest path, 334-335
- Single-source shortest paths, 73, 127, 281
 - acyclic networks, 314-315
 - Dijkstra's algorithm, 293-294
 - and mincost flow problem, 472-473
 - negative weights, 350-354

- reachability, 166-167
- Sink, 154, 197, 281, 375-378
- Software library, 485
- Sollin, G., 267
- Sorting, 331
- Source vertex, 14, 154, 197, 281, 375-378
- Source-sink shortest path, 281, 294, 322-326
- Spanning forest, 12, 92-94, 109
- Spanning tree, 12, 72-73
 - feasible, 453-455
 - maxflow, 454-455
 - network simplex algorithm, 457-464
 - see also* Minimum spanning tree
- Sparse graph, 13, 25, 29-30
- st*-connectivity, 119
- st*-cut, 385-387, 427-428
- st*-network, 375-378, 429-430
- Stack, LIFO, 132-134
- Stack-based augmenting-path method, 400, 402
- Static graph, 42, 50
- Strong components, 157-158, 205-215
 - transitive closure, 217-219
- Strong connectivity, 72, 156-158, 205-206
- Subgraph, 9
 - forbidden, 73
 - maximal connected, 12
- Subset inclusion, 184
- Supply lines problem, 370, 386-387
- Symbol table
 - find edge, 41
 - graph-building function, 51
 - indexing function, 52
- Symmetric relation, 183

- Tarjan, R. E., 410
- Tarjan's algorithm, 73, 207, 210-212
- Telephone network, 370

- Topological sorting, 150, 187, 191-200
 - DFS, 193-197, 202-203
 - multisource shortest paths, 314-318
 - relabeling/rearrangement, 192
 - reverse, 193-197
- Total order, 185
- Tour, 10
 - Euler, 62-68, 108-109
 - Hamilton, 60-62
 - postman problem, 75
 - traveling salesperson problem, 76
- Tractable problem, 73
- Traffic flow, 369
- Transaction, 5
- Transaction graph, 50
- Transitive closure, 72, 169-180
 - abstract, 174-175, 217-220
 - and all-pairs shortest paths, 175, 328-329
 - Boolean matrix multiplication, 169-172, 176-177
 - DAGs, 202-204
 - DFS-based, 177-179
 - performance, 172-174, 177, 179-180, 222
 - of a relation, 183
 - Warshall's algorithm, 172-175
- Transitive reduction, 224
- Transitive relation, 183
- Transportation problem, 368, 475-476
- Traveling salesperson problem, 76
- Tree, 12, 72-73
 - BFS, 123-129
 - binary, 189-190
 - and DAGs, 189-190
 - DFS, 90, 97-103, 113-116, 161-162
 - edge, 99, 113-114, 161-162, 202
 - preorder tree traversal, 100
 - shortest paths tree (SPT), 279, 287-288, 294
 - spanning, 453-455, 457-464
 - see also* Minimum spanning tree (MST)
- Tree link, 99-100
- Trémaux exploration, 83-90, 108-109
- Two-coloring, 109-110
- Two-way Euler tour, 68, 108-109

- Uncapacitated edge, 381
- Undirected graph, 14-15, 37
 - and digraphs, 153, 155-162, 164-167, 179-180
 - networks, 429-430
 - reachability, 205
 - underlying, 15
- Unconnected digraph, 224
- Union, 12
- Union-find algorithm, 24, 44
 - Boruvka's algorithm, 265
 - and DFS, 107-108
 - Kruskal's algorithm, 263
 - performance, 146

- Vector, vertex-indexed, 96
- Vertex, 7-10
 - active, 411
 - adjacent, 9
 - connectivity, 118-119, 438
 - cut, 117-118
 - degree of, 9
 - destination, 14
 - disjoint, 11
 - dummy, 376-377
 - excess, 411
 - fringe, 253
 - height, 412-415
 - indegree/outdegree, 14
 - inflow/outflow, 375-378
 - marking, 92
 - ordered pair, 14
 - potentials, 416-419, 454-456, 468
 - reachable, 152, 158
 - removing, 41-42

- search, 109
- separation, 117-118
- sink/source, 14, 154, 197, 281, 375-378
- Vertex-based preflow-push algorithm, 416
- Vertex-indexed array, 37-38
- Vertex-indexed vector, 96
- V-vertex graph, 7

- Warshall's algorithm, 172-175, 289-290, 307-309
- Weighted diameter, 304
- Weighted graph, 15, 227-229
 - adjacency-matrix representation, 37
 - ADTs, 230-239
 - arbitrary weights, 228
 - bipartite matching, 74
 - digraphs, 277-278
 - edge weights, 232-234, 482
 - equal weights, 228-229
 - integer weights, 379-380
 - negative weights, 345-365
 - path weight, 277
 - reweighting, 324, 356-360
 - sentinel weight, 234-235, 278
 - see also* Minimum spanning tree (MST); Shortest path
- Whitney's theorem, 119
- World Wide Web, 4
- Worst-case performance, 43-44