Index

80/20 rule, 309, 333

abstract data type (ADT), xiv, 8–12, 20, 47, 93–97, 131–138, 149, 163, 196–198, 206, 207, 216, 217, 277–282, 371, 376, 378, 413, 428, 456
abstraction, 10
accounting, 117, 125
Ackermann's function, 215
activation record, *see* compiler, activation record
aggregate type, 8
algorithm analysis, xiii, 4, 53–89, 223
  amortized, *see* amortized analysis
  asymptotic, 4, 53, 54, 63–68, 93, 461
  empirical comparison, 53–54, 83, 224
  for program statements, 69–73
  multiple parameters, 77–78
  running time measures, 55
  space requirements, 54, 78–80
algorithm, definition of, 17–18
all-pairs shortest paths, 513–515, 532, 535
amortized analysis, 71, 111, 311, 461, 476–477, 479, 481, 482
approximation, 553
array
  dynamic, 111, 481
  implementation, 8, 9, 20
artificial intelligence, 371
`assert`, xvii
asymptotic analysis, *see* algorithm analysis, asymptotic
ATM machine, 6
average-case analysis, 59–60
AVL tree, 188, 349, 429, 434–438, 456

back of the envelope, napkin, *see* estimating
backtracking, 553
bag, 24, 47
bank, 6–7
basic operation, 5, 6, 20, 55, 56, 61
best fit, *see* memory management, best fit
best-case analysis, 59–60
big-Oh notation, *see* O notation
bin packing, 554
binary search, *see* search, binary
binary search tree, *see* BST
binary tree, 145–195
  BST, *see* BST
  complete, 146, 147, 161, 162, 171, 243
  full, 146–149, 160, 179, 189, 214
  implementation, 145, 147, 188
  node, 145, 149, 154–158
  `null` pointers, 149
  overhead, 160

parent pointer, 154
space requirements, 148, 154, 160–161
terminology, 145–147
threaded, 192
traversal, *see* traversal, binary tree
Binsort, 79, 80, 244–251, 254, 321, 538
bintree, 451, 456
birthday problem, 315, 337
block, 283, 288
Boolean expression, 547
clause, 547
Conjunctive Normal Form, 547
literal, 547
Boolean variable, 8, 28, 89
branch and bounds, 553
breadth-first search, 371, 384, 386, 387, 400
BST, xv, 163–170, 188, 190, 193, 219, 237, 243, 348–353, 358, 429, 434–440, 442, 451, 456, 516, 522
efficiency, 169
insert, 164–167
remove, 167–169
search, 163–164
search tree property, 163
traversal, *see* traversal
B-tree, 302, 314, 342, 346, 355–365, 367, 453
analysis, 364–365
B^+-tree, 8, 10, 342, 346, 358–365, 367, 368, 430
B^*-tree, 362
Bubble Sort, 74, 227–228, 230–231, 252, 258
buffer pool, xv, 11, 265, 274–282, 298, 299, 309, 310, 348, 357, 421
ADT, 277–282
replacement schemes, 275–276, 310–312
cache, 268, 274–282, 309
CD-ROM, 9, 266, 269, 315
ceiling function, 28
city database, 142, 193, 446, 456
class, *see* object-oriented programming, class
clique, 545, 550–552, 563
cluster problem, 219, 456
cluster, file, 270, 273, 420
code tuning, 53–55, 81–84, 242–243
Collatz sequence, 67, 87, 555, 562, 565
compiler, 83
activation record, 121
efficiency, 54
optimization, 82
complexity, 10
composite, *see* design pattern, composite
composite type, 8
computability, 19, 555, 562
computer graphics, 430, 440
connected component, *see* graph, connected component
contradiction, proof by, *see* proof, contradiction
cost, 5
cylinder, *see* disk drive, cylinder

data item, 8
data member, 9
data structure, 4, 9
costs and benefits, xiii, 3, 6–8
definition, 9
philosophy, 4–6
physical vs. logical form, xv, 8–9, 11–12, 93, 172, 268, 278, 405
selecting, 5–6
spatial, *see* spatial data structure
data type, 8
decision problem, 544, 548, 563
decision tree, 254–257, 490–491
decomposition

image space, 430
key space, 430
object space, 429
depth-first search, 371, 383–385, 400, 424, 482
deque, 141
dequeue, *see* queue, dequeue
design pattern, xiv, 12–16, 19
composite, 14–15, 158, 451
flyweight, 13, 158, 192, 450–451
strategy, 15–16, 138
visitor, 13–14, 152, 383, 402
Deutsch-Schorr-Waite algorithm, 425, 428
dictionary, xiv, 163, 329, 431
ADT, 131–137, 301, 339, 368, 509
Dijkstra's algorithm, 390–394, 400, 401, 514
Diminishing Increment Sort, *see* Shellsort
directed acyclic graph (DAG), 373, 384, 400, 406, 424
discrete mathematics, xiv, 45
disjoint, 145
disjoint set, *see* equivalence class
disk drive, 9, 265, 268–297
access cost, 272–274, 295
cylinder, 269, 347
organization, 268–271
disk processing, *see* file processing
divide and conquer, 237, 240, 242, 304, 467, 472–474
document retrieval, 314, 335
double buffering, 275, 287, 288
dynamic array, *see* array, dynamic
dynamic memory allocation, 100
dynamic programming, 509–515, 532, 553

efficiency, xiii, 3–5
element, 23
homogeneity, 94, 112
implementation, 111–112
Emacs text editor, 423, 425
encapsulation, 9
enqueue, *see* queue, enqueue
entry-sequenced file, 341
enumeration, *see* traversal
equation, representation, 155
equivalence, 25–26
class, 25, 195, 200–206, 215, 216, 219, 397, 398, 401, 403, 456
relation, 25, 46
estimation, 23, 44–46, 50, 51, 53–55, 63
exact-match query, *see* search, exact-match query
exponential growth rate, *see* growth rate, exponential
expression tree, 154–158
extent, 271
external sorting, *see* sorting, external

factorial function, 27, 32, 34, 43, 47, 71, 79, 85, 123, 254, 257, 562
Stirling's approximation, 27, 257
Fibonacci sequence, 32, 47–49, 89, 469–470, 509
FIFO list, 125
file access, 282–283
file manager, 268, 270, 274, 414, 415, 421
file processing, 80, 224, 295
file structure, 9, 267, 341, 365
first fit, *see* memory management, first fit
floor function, 28
floppy disk drive, 269
Floyd's algorithm, 513–515, 532, 535
flyweight, *see* design pattern, flyweight
fragmentation, 271, 274, 415, 419–421
external, 415
internal, 271, 415
free store, 107–108

free tree, 373, 393, 399
freelist, 117, 120
full binary tree theorem, 147–149, 160, 189, 213
function, mathematical, 16

garbage collection, 106
general tree, 195–219
  ADT, 196–197, 216
  converting to binary tree, 210, 217
  dynamic implementations, 217
  implementation, 206–210
  left-child/right-sibling, 206, 207, 217
  list of children, 206, 217, 373
  parent pointer implementation, 199–206, 437
  terminology, 195–196
  traversal, *see* traversal
generics, xvi, 12, 95
Geographic Information System, 7–8
geometric distribution, 308, 519, 522
gigabyte, 27
graph, xv, 22, 371–403, 407
  adjacency list, 371, 373, 374, 381, 400
  adjacency matrix, 371, 373, 374, 378, 379, 400, 408
  ADT, 371, 376, 378
  connected component, 373, 402, 482
  edge, 372
  implementation, 371, 376–378
  modeling of problems, 371, 380, 384, 389, 390, 393
  parallel edge, 372
  representation, 373–376
  self loop, 372
  terminology, 371–373
  traversal, *see* traversal, graph
  undirected, 372, 408
  vertex, 372
greatest common divisor, *see* largest common factor
greedy algorithm, 183, 396, 397
growth rate, 53, 56–58, 85
  asymptotic, 63
  constant, 56, 64
  exponential, 58, 62, 536, 541–555
  linear, 58, 61, 63, 80
  quadratic, 58, 61, 62, 80, 81

halting problem, 555–561
Hamiltonian cycle, 563
Harmonic Series, 31, 309, 475
hashing, 7, 10, 29, 60, 302, 314–335, 341, 342, 355, 409, 453, 481
  analysis of, 331–334
  bucket, 321–323, 339
  closed, 320–329, 338
  collision resolution, 315, 321–329, 334, 335
  deletion, 334–335, 338
  double, 329, 338
  dynamic, 335
  hash function, 315–320, 337
  home position, 321
  linear probing, 324–326, 329, 333, 334, 337, 339
  load factor, 331
  open, 320–321
  perfect, 315, 335
  primary clustering, 326–329
  probe function, 324, 325, 327–329
  probe sequence, 324–329, 331–335
  pseudo-random probing, 327, 329
  quadratic probing, 328, 329, 337, 338
  search, 324
  table, 314
  tombstone, 334
header node, 120, 128
heap, 145, 147, 161, 170–177, 188, 191, 193, 243–244, 263, 391, 397

building, 172–177
for memory management, 414
insert, 172
max-heap, 171
min-heap, 171, 289
partial ordering property, 171
remove, 177
siftdown, 176, 177, 263
Heapsort, 171, 243–245, 252, 263, 288
heuristic, 553–554
hidden obligations, *see* obligations, hidden
Huffman coding tree, 145, 147, 154, 178–188, 191–194, 218, 430
prefix property, 186

independent set, 563
index, 11, 259, 341–368
file, 285, 341
inverted list, 345, 366
linear, 8, 343–345, 366, 367
tree, 342, 348–365
induction, 217
induction, proof by, *see* proof, induction
inheritance, xvi, 95, 97, 103, 137, 156, 157, 159, 161
inorder traversal, *see* traversal, inorder
input size, 55, 59
Insertion Sort, 74, 225–228, 230–233, 242, 252, 254–259, 261, 262
Double, 262
integer representation, 4, 8–9, 20, 142
inversion, 227, 231
inverted list, *see* index, inverted list
ISAM, 342, 346–348, 366

k-d tree, 442–447, 451, 453, 456
K-ary tree, 210–211, 215, 217, 447
key, 131–133
kilobyte, 27
knapsack problem, 552, 564, 565
Kruskal's algorithm, xv, 244, 397–398, 401

largest common factor, 49, 523–524
latency, 270, 272, 273
least frequently used (LFU), 276, 298, 310
least recently used (LRU), 276, 298, 299, 310, 357
LIFO list, 117
linear growth, *see* growth rate, linear
linear index, *see* index, linear
linear search, *see* search, sequential
link, *see* list, link class
linked list, *see* list, linked
LISP, 46, 407, 422, 424, 425
list, 22, 93–143, 145, 179, 342, 405, 407, 481
ADT, 9, 93–97, 138
append, 99, 113, 114
array-based, 8, 93, 97–100, 108–111, 117, 142
basic operations, 94
circular, 140
comparison of space requirements, 140
current position, 94, 95, 102–103, 106, 111
doubly linked, 112–117, 140, 142, 154
space, 114–117
element, 94, 111–112
freelist, 107–108, 414–424
head, 94, 100, 102
implementations compared, 108–111
initialization, 94, 97, 99
insert, 94, 99, 100, 102–106, 111, 113–114, 116, 145
link class, 100–101
linked, 8, 93, 97, 100–111, 344, 405, 429, 516, 517, 521

node, 100–103, 113, 114
notation, 94
ordered by frequency, 307–313, 461
orthogonal, 410
remove, 94, 99, 103, 106, 107, 111, 113, 114, 116
search, 145, 301–313
self-organizing, xv, 60, 310–313, 334–337, 339, 437, 478–479
singly linked, 101, 112, 113
sorted, 4, 94, 137
space requirements, 108–110, 140
tail, 94
terminology, 94
unsorted, 94
locality of reference, 270, 274, 332, 348, 355
logarithm, 29–30, 47, 527–528
log*, 205, 215
logical representation, *see* data structure, physical vs. logical form
lookup table, 79
lower bound, 53, 65–68, 332
sorting, 253–257, 538

map, 371, 388
Master Theorem, *see* recurrence relation, Master Theorem
matching, 553
matrix, 408–412
multiplication, 540, 541
sparse, xv, 9, 405, 409–412, 426, 427
triangular, 408, 409
megabyte, 27
member, *see* object-oriented programming, member
member function, *see* object-oriented programming, member function
memory management, 11, 405, 412–427
ADT, 413, 428
best fit, 418, 554
buddy method, 415, 419–420, 428
failure policy, 415, 421–425
first fit, 418, 554
garbage collection, 422–425
memory allocation, 413
memory pool, 413
sequential fit, 415–419, 427
worst fit, 418
Mergesort, 123, 233–236, 248, 252, 261, 467, 472, 474
external, 286–288
multiway merging, 290–294, 298, 300
metaphor, 10, 19
Microsoft Windows, 270, 295
millisecond, 27
minimum-cost spanning tree, 223, 244, 371, 393–398, 401, 535
modulus function, 26, 28
move-to-front, 310–313, 337, 478–479
multilist, 24, 405–408, 426
multiway merging, *see* Mergesort, multiway merging

nested parentheses, 21–22, 141
networks, 371, 390
`new`, 107–108, 113, 114, 414
$\mathcal{NP}$, *see* problem, $\mathcal{NP}$
null pointer, 101

O notation, 63–68, 85
object-oriented programming, 9, 11–16, 19–20
class, 9, 94
class hierarchy, 14–15, 154–158, 450–451
members and objects, 8, 9
obligations, hidden, 138, 152, 281

octree, 451
Ω notation, 65–68, 85
one-way list, 101
operating system, 18, 170, 268, 270, 273–276, 285, 288, 414, 421, 426
overhead, 79, 108–110
  binary tree, 190
  matrix, 410
  stack, 121

pairing, 536–538
palindrome, 140
partial order, 26, 47, 171
  poset, 26
partition, 563
path compression, 204–206, 216, 483
permutation, 27, 47, 48, 79, 80, 244, 255–257, 331
physical representation, *see* data structure, physical vs. logical form
Pigeonhole Principle, 49, 50, 128
point quadtree, 452, 456
pop, *see* stack, pop
postorder traversal, *see* traversal, postorder
powerset, *see* set, powerset
PR quadtree, 13, 154, 210, 442, 447–451, 453, 455, 456
preorder traversal, *see* traversal, preorder
prerequisite problem, 371
Prim's algorithm, 393–396, 401
primary index, 342
primary key, 342
priority queue, 145, 161, 179, 193, 391, 394
probabilistic data structure, 509, 516–522
problem, 6, 16–18, 536
  analysis of, 53, 74–75, 224, 253–257
  hard, 541–555
  impossible, 555–561
  instance, 16
  $\mathcal{NP}$, 543
  $\mathcal{NP}$-complete, 543–555, 561
  $\mathcal{NP}$-hard, 545
problem solving, 19
program, 3, 18
  running time, 54–55
programming style, 19
proof
  contradiction, 37–38, 49, 50, 396, 538, 559, 560
  direct, 37
  induction, 32, 37–43, 46, 49, 50, 148, 176, 184–185, 189, 257, 399, 462–465, 467, 468, 471, 480, 481, 483, 501
pseudo-polynomial time algorithm, 552
pseudocode, xvii, 18
push, *see* stack, push

quadratic growth, *see* growth rate, quadratic
queue, 93, 125–131, 140, 141, 384, 387, 388
  array-based, 125–128
  circular, 126–128, 140
  dequeue, 125, 126, 128
  empty vs. full, 127–128
  enqueue, 125, 126
  implementations compared, 131
  linked, 128, 130
  priority, *see* priority queue
  terminology, 125
Quicksort, 123, 227, 236–244, 252, 259, 262, 284–286, 288, 336, 461
  analysis, 474–475

Radix Sort, 247–252, 254, 263
RAM, 266, 267
Random, 28
range query, 342, *see* search, range query
real-time applications, 59, 60
recurrence relation, 32–33, 50, 241, 461, 467–475, 479, 480
  divide and conquer, 472–474
  estimating, 467–470
  expanding, 470, 472, 481
  Master Theorem, 472–474
  solution, 33
recursion, xiv, 32, 34–36, 38, 39, 47–49, 71, 122, 150–152, 164, 189, 193, 234–236, 259, 261, 262, 424
  implemented by stack, 121–125, 242
  replaced by iteration, 48, 123
reduction, 254, 536–541, 560, 562, 564
relation, 25–27, 46, 47
replacement selection, 171, 288–290, 293, 298, 300, 461
resource constraints, 5, 6, 16, 53, 54
run (in sorting), 286
run file, 286, 287
running-time equation, 56

satisfiability, 547–553
Scheme, 46
search, 21, 80, 301–339, 341
  binary, 29, 71–73, 87–89, 258, 304, 336, 343, 357, 472, 481
  defined, 301
  exact-match query, 7–8, 10, 301, 302, 341
  in a dictionary, 304
  interpolation, 304–307, 336
  jump, 303–304
  methods, 301
  multi-dimensional, 440
  range query, 7–8, 10, 301, 314, 341, 348
  sequential, 21, 55–56, 59–60, 64, 65, 71–73, 88, 89, 302–303, 312, 336, 476
  sets, 313–314
  successful, 301
  unsuccessful, 301, 331
search trees, 60, 170, 342, 346, 349, 355, 434, 437, 440
secondary index, 342
secondary key, 342
secondary storage, 265–274, 295–298
sector, 269, 271, 273, 284
seek, 269, 272
Selection Sort, 229–231, 242, 252, 258
self-organizing lists, *see* list, self-organizing
sequence, 25, 27, 47, 94, 302, 313, 343, 536
sequential search, *see* search, sequential
sequential tree implementations, 212–215, 217, 218
serialization, 212
set, 23–27, 47
  powerset, 24, 27
  search, 302, 313–314
  subset, superset, 24
  terminology, 23–24
  union, intersection, difference, 24, 313, 337
Shellsort, 227, 231–233, 252, 259
shortest paths, 371, 388–393, 400
simulation, 83
Skip List, xv, 516–522, 532, 533
slide rule, 30, 527
software engineering, xiii, 4, 19, 536
sorting, 17, 21, 22, 55, 59, 60, 74–75, 77, 80, 223–263, 303, 312, 536–538
  adaptive, 257

comparing algorithms, 224–225, 251–253, 294
exchange sorting, 230–231
external, xv, 161, 224, 243, 265, 283–295, 298–300
internal, xv
lower bound, 224, 253–257
small data sets, 225, 242, 257, 260
stable algorithms, 224, 258
terminology, 224–225
spatial data structure, 429, 440–453
splay tree, 170, 188, 349, 429, 434, 437–440, 453, 454, 456, 461, 517
stable sorting alorithms, *see* sorting, stable algorithms
stack, 93, 117–125, 140, 141, 189, 193, 242, 258, 259, 262, 383–385, 477
array-based, 117–118
constructor, 117
implementations compared, 121
insert, 117
linked, 120
pop, 117, 118, 120, 142
push, 117, 118, 120, 142
remove, 117
terminology, 117
top, 117–118, 120
two in one array, 121, 140
variable-size elements, 142
Strassen's algorithm, 524, 533
strategy, *see* design pattern, strategy
subclass, *see* object-oriented programming, class hierarchy
subset, *see* set, subset
suffix tree, 455
summation, 30–32, 39, 40, 49, 50, 70, 71, 88, 170, 177, 240, 241, 308, 309, 409, 461–466, 471, 473, 474, 476, 477, 479
guess and test, 479
list of solutions, 31, 32
notation, 30
shifting method, 463–466, 475, 480
`swap`, 28

tape drive, 268, 283
text compression, 145, 178–188, 312–313, 335, 339
Θ notation, 66–68, 87
topological sort, 371, 384–388, 400
total order, 26, 47, 171
Towers of Hanoi, 34–36, 123, 535, 542
tradeoff, xiii, 3, 13, 73, 271, 283
disk-based space/time principle, 80, 332
space/time principle, 79–80, 95, 115, 178, 333
transportation network, 371, 388
transpose, 311, 312, 337
traveling salesman, 543–545, 552, 553, 563, 565
traversal
binary tree, 123, 145, 149–153, 158, 163, 170, 189, 380
enumeration, 149, 163, 212
general tree, 197–198, 216
graph, 371, 380–388
tree
height balanced, 354, 355, 357, 517
terminology, 145
trie, 154, 178, 188, 251, 429–434, 454, 455
alphabet, 430
binary, 430
PATRICIA, 431–434, 453, 454
tuple, 25
Turing machine, 548
two-coloring, 42
2-3 tree, 170, 342, 350–354, 357, 360, 367, 368, 434, 481, 516

type, 8

uncountability, 556–558
UNION/FIND, xv, 199, 398, 403, 461, 483
units of measure, 27, 46
UNIX, 237, 270, 295, 423
upper bound, 53, 63–67

variable-length record, 142, 343, 367, 405, 407, 412
  sorting, 225
vector, 25, 111
vertex cover, 549, 550, 552, 553, 563, 564
virtual function, 161
virtual memory, 276–278, 285, 298
visitor, *see* design pattern, visitor

weighted union rule, 204, 216, 483
worst fit, *see* memory management, worst fit
worst-case analysis, 59–60, 65

Zipf distribution, 309, 316, 336
Ziv-Lempel coding, 313, 335