

Index

A

algorithms, 7
 adapting, 55
 parallel, 110–112, 115–116
 recursive, 155
 search. *See* search algorithms
amortized cost, 172
ArrayCart, 21
arrays, 15, 19–24
 and binary search, 36–38
 for heaps, 219–222
 index of, 19, 24, 28
 for queues, 100–101
 sorted, 36
 and binary search tree
 construction, 147–150
 for stacks, 99–100
 for strings, 25–28

B

back pointers, 80
backtracking, 67–70,
 84–88, 106
best-first search, 189–192
 for investigations, 189
 for locks, 203–205
 priority queues, 196,
 203–205

binary search, 33–38, 41–43,
 49–52

binary search ladders,
 138–142, 145–149
 for artwork, 138, 147–149
 snake rungs, 147, 149
 as traps, 139–140

binary search trees, 138–144
 balanced, 144, 168
 child node, 138, 143
 construction, 147–150
 inserting nodes, 167–169
 property, 139, 143, 171, 173
 range search, 153–162
 root node, 138, 143
 searching, 139–144
 subtree, 138, 143
 unbalanced tree, 168–169

Boolean (citizen of Bool),
 11, 223

breadth-first search,
 72–81, 121
 on graphs, 77–81
 on grids, 84–85
 for locks, 72–77
 and queues, 77, 107

C

characters, in strings, 25, 28
child nodes, 138, 143

D

- data structures
 - arrays, 15, 19–24
 - binary search trees, 138–144
 - graphs, 77–78, 91, 108, 136–137
 - grids, 84
 - heaps, 214–217, 219–222
 - inverted indexes, 129–133, 226
 - priority queues, 194–197, 199–200, 203–205
 - queues, 77, 96–97, 100–101, 107, 195
 - stacks, 91, 94–96, 99–100, 195
 - strings, 25–28
 - trees, 73, 138–144
 - tries, 177–182
- depth-first search, 85–88, 91–92, 155
 - on graphs, 91–92, 136–137
 - on grids, 85–88
 - and iterative deepening, 120–123, 125–126
 - and stacks, 91, 105–106

E

- exhaustive search, 10–13, 15–16, 43, 226
- edge, of graph, 78, 91, 136

F

- first-in, first-out data structures, 96–97, 100. *See also* queues

G

- graphs, 77–78, 108
 - and breadth-first search, 77–81
 - cost of edge, 80–81
 - and depth-first search, 91–92, 136–137
 - edge, 78, 91, 136
 - and iterative deepening, 125–126
 - and maps, 78, 91, 136–137
 - neighbor, 78
 - node, 78, 91, 136
- grids, 84
 - and breadth-first search, 84–85
 - and depth-first search, 85–88
 - and iterative deepening, 120–123

H

- heaps, 214–217
 - and arrays, 219–222
 - inserting element, 220–221
 - maximum element, 220
 - priority queue, 219
 - property, 219
 - removing max element, 221–222
 - root node, 220
- heuristics, 209–212

I

- index
 - in arrays 19, 24, 28
 - inverted, 129–133, 226

intersection, 130
inverted indexes,
 129–133, 226
iterative deepening, 118–126
 on graphs, 125–126
 on grids, 120–123

K

Kingdom Highway Map,
 78, 91

L

last-in, first-out data structures, 95–96, 99. *See also* stacks

League of Magical
 Confectioners, 151

linear search, 15–16

locks,
 and best-first search,
 203–205
 and breadth-first search,
 72–77
 and priority queues,
 203–205

M

Manhattan distance, 191

N

nodes
 child, 138, 143
 graph, 78, 91, 136
 inserting
 in binary search tree,
 167–169
 in heap, 220–221

root
 of binary search tree,
 138, 143
 of heap, 220
tree, 73, 138, 143–144

P

parallel algorithms, 110–112,
 115–116
prefix trees, 177–182
priority queues, 194–197,
 199–200
 and best-first search, 196,
 203–205
 dequeue, 195
 enqueue, 195
 and heaps, 219
 for investigations, 194–197
 for locks, 203–205

Q

queues, 77, 96–97, 195. *See also* priority queues
 and arrays, 100–101
 and breadth-first search, 107
 dequeue, 97, 100–101
 enqueue, 97, 100

R

range search, 153–162
recursive algorithms, 155
 for binary search tree construction, 147–150
 for range search, 153–162
root node
 of binary search tree,
 138, 143
 of heap, 220

S

search algorithms
 binary search, 33–38,
 41–43, 49–52
 best-first search. *See* best-
 first search
 breadth-first search. *See*
 breadth-first search
 depth-first search. *See*
 depth-first search
 exhaustive search, 10–13,
 15–16, 43, 226
 linear search, 15–16
 range search, 153–162
search space, 5–7
 and binary search, 36
 and grids, 84
search state, 5–7
 and backtracking, 69–70
 on graphs, 78
 on grids, 84
set intersection, 130
sorted data, 33, 36, 43,
 45–46

stacks, 91, 94–96, 195
 and arrays, 99–100
 and depth-first search, 91,
 105–106
 pop, 95, 99–100
 push, 95, 99
state. *See* search state
strings, 25–28

T

target, 5, 7
trees, 73, 138–144
 binary search. *See* binary
 search trees
 heaps. *See* heaps
 nodes, 73, 138, 143–144
 child, 138, 143
 root, 138, 143
 subtrees, 138, 143
tries, 177–182

W

Weather Records problem, 43