Index

Data.Array, 2, 25, 29, 100, 114, 123
Data.Array.ST, 3
Data.Map, 271
Data.Sequence, 114
Data.Set, 70
Either, 168
Ix, 2
Queue, 115, 249
QuickCheck, 188, 196
Ratio, 182, 199
↓ – after, 128
↑ – before, 109
\ – list difference, 1, 3, 22, 59, 64, 142, 192
↑ – exponentiation, 183
∧ – merge, 9, 10, 29
!! – list index, 87, 93, 100
! – array index, 25, 29, 87, 100
≤ – prefix, 103, 119, 127
accumArray, 2, 5, 82, 123
applyUntil, 82
array, 29, 85
bounds, 25
break, 154, 164, 182
cmpare, 29
concatMap, 42
elems, 85
foldr – fold over nonempty lists, 42
fork, 35, 83, 94, 118
init, 66, 67, 117
listArray, 25, 100
minors, 172
nodups, 149
nub, 64
partition, 4
partitions, 38
reverse, 119, 244
scanl, 118, 238
scanr, 70
sort, 28, 95
sortBy, 29, 94
span, 67
subseqs, 57, 65, 157, 163
tails, 7, 79, 100, 102
transpose, 98, 150, 193
unfodlr, 202, 243
zip, 35, 83
zipWith, 83
Abelian group, 27
abides property, 3, 22
abstraction function, 129, 211, 226
accumulating function, 2
accumulating parameter, 131, 138, 140, 177, 253
adaptive encoding, 200
amortised time, 5, 118, 131, 133
annotating a tree, 170
arithmetic decoding, 201
arithmetic expressions, 37, 156
array update operation, 3, 6
arrays, 1, 2, 21, 29, 85, 99
association list, 29, 238
asymptotic complexity, 27
bags, 25, 50, 51
balanced trees, 21, 54, 234
Bareiss algorithm, 186
bijection, 129
binary search, 7, 10, 14, 15, 19, 54
binomial trees, 178
bioinformatics, 77, 90
Boolean satisfiability, 155
borders of a list, 103
bottom-up algorithm, 41
boustrophedon product, 245, 251, 260
breath-first search, 136, 137, 178
Bulldozer algorithm, 196
bzip2, 101
call-tree, 168
Cartesian coordinates, 141, 155
Cartesian product, 149
celebrity clique, 56
Chio’s identity, 182
clique, 56
combinatorial patterns, 242
comparison-based sorting, 10, 16, 27
computional geometry, 188
conjugate, 263
constraint satisfaction, 155
continuations, 273
coroutines, 273
Index

cost function, 41, 48, 52
cyclic structures, 133, 179
data compression, 91, 198
data mining, 77
data refinement, 5, 48, 108, 114, 129, 210
deforestation, 168
depth-first search, 137, 221, 222
destreaming, 214
destreaming theorem, 214
Dilworth’s theorem, 54
divide and conquer, 1, 3, 5, 7, 8, 15, 21–23,
27, 29, 30, 65, 81, 171
dot product, 185
dynamic programming, 168

EOF (end-of-file symbol), 203
exhaustive search, 12, 35, 39, 57, 148, 156

facets, 190
failure function, 133
fictitious values, 14, 77
finite automaton, 74, 136
fission law of foldl, 130
fixpoint induction, 205
forests, 42, 174
fringe of a tree, 41
frontier, 137
fully strict composition, 243
fusion law of foldl, 76, 130, 195
fusion law of folddr, 34, 51, 52, 61, 247, 260,
261, 265
fusion law of foldr, 43
fusion law of fork, 35
fusion law of unfoldr, 206, 212

Galil’s algorithm, 122
garbage collection, 165, 166
Garsia–Wachs algorithm, 49
Gaussian elimination, 180
graph traversal, 178, 221
Gray path order, 258
Greedy algorithms, 41, 48, 50, 140
Guilfield’s Z algorithm, 116

Hu–Tucker algorithm, 49
Huffman coding, 91, 198, 201

immutable arrays, 25
incremental algorithm, 188, 191, 204
incremental decoding, 216
incremental encoding, 203, 209
indexitis, 150
inductive algorithm, 42, 93, 102
integer arithmetic, 182, 198, 208
integer division, 182
intermediate data structure, 168
interval expansion, 209, 210
inversion table, 10
inverting a function, 12, 93
involution, 150
iterative algorithm, 10, 82, 109, 113

Knuth and Ruskey algorithm, 258
Knuth’s spider spinning algorithm, 242
Koda–Ruskey algorithm, 242
law of iterate, 99
laws of filter, 118, 152
laws of fork, 35
lazy evaluation, 33, 147, 185, 243
leaf-labelled trees, 41, 165, 168
left spines, 43, 45, 177
left-inverse, 129
Leibniz formula, 180
lexicographic ordering, 45, 52, 64, 102, 104
linear ordering, 43
linked list, 225
longest common prefix, 103, 112, 120
longest decreasing subsequence, 54
loop invariants, 62, 111
lower bounds, 16, 27, 28, 64

Mahajan and Vinay’s algorithm, 186
majority voting problem, 62
matrices, 147, 181
matrix Cartesian product, 149
maximum marking problems, 77
maximum non-segment sum, 73
maximum segment sum, 73
maximum surpasser count, 7
McCarthy S-expression, 221
memo table, 163
memoisation, 162
merge, 26, 142, 158
mergesort, 29, 89, 171, 173
minimal element, 53
minimum cost tree, 44
minimum element, 53
minors, 181
model checking, 155
monads, 3, 114, 155
monotonicity condition, 48, 53
move-to-front encoding, 91
multisets, 25
narrowing, 199
non-deterministic functions, 43, 51
normal form, 160

online list labelling, 241
Open Problems Project, 31
optimal bracketing, 176
optimisation problems, 48, 176
order-maintenance problem, 241
overflow, 214

parametricity, 62
partial evaluation, 134
partial ordering, 53
partial preorder, 52
partition sort, 85
partition sorting, 87
perfect binary trees, 171

© in this web service Cambridge University Press
www.cambridge.org
Index

permutations, 79, 90, 91, 96, 97, 180, 189, 242, 251
planning algorithm, 136, 138
plumbing combinators, 36
prefix, 66
prefix ordering, 103, 105, 119
preorder traversal, 245, 270
principal submatrices, 185
program transformation, 221
PSPACE completeness, 136
queues, 109, 137, 248, 249
Quicksort, 5, 85, 89
radix sort, 95, 101
ranking a list, 79
rational arithmetic, 180, 188, 198
rational division, 181
recurrence relations, 15, 31, 88
refinement, 44, 48, 51–53, 80
regular cost function, 49
regular expression, 74
relations, 48, 167, 229
representation function, 129, 211
right spines, 177
Rose trees, 164, 245
rotations of a list, 91
rule of floors, 215
run-length encoding, 91
sadleback search, 14
safe replacement, 222
scan lemma, 118, 125
segments, 73, 171
Shannon–Fano coding, 198
sharing, 168, 173
shortest upravel, 50
simpex, 188
skeleton trees, 165
sliding-block puzzle, 136
smart constructors, 48, 170, 177
smooth algorithms, 241
solving a recursion, 98
sorting, 9, 10, 16, 91, 149
sorting numbers, 1, 3
sorting permutation, 10
space/time trade-offs, 156
spanning tree, 178
stable sorting algorithm, 86, 95
stacks, 137, 221, 222
streaming, 203, 214
streaming theorem, 204
string matching, 112, 117, 127
stringology, 103
subsequences, 50, 64, 74, 162, 177, 242
suffix tree, 101
suffixes, 79, 100
Sylvester’s identity, 186
thinning algorithm, 161
top-down algorithm, 41
totally acyclic digraph, 258
transitions, 242
trees, 130, 165, 248
tries, 163
tupling law of foldl, 118, 125
tupling law of foldr, 247
unfolds, 168
unmerges, 158, 159, 165
unravel, 50
upper triangular matrix, 185
Vandermonde’s convolution, 17
well-founded recursion, 4, 30
while loop, 111, 113
wholemeal programming, 150
windows of a text, 120
Young tableau, 28