Index

$-notation, 31–34

abstract data type, 3
abstraction, 3
accumulated debt, 58, 60
accumulated savings, 40, 58
accumulating parameter, 86, 95
actual cost, 60
AltBinaryRandomAccessList (structure), 147
amortization
  banker's method, *see* banker's method
  eliminating, *see* scheduling
  physicist's method, *see* physicist's method
  problem with persistence, 54–55
  traditional, 39–41
anticredits, 81
assignments, 2

banker's method
  justification of, 62–63
  traditional, 40
  with lazy evaluation, 61–62, 69
BankersDeque (functor), 110
BankersQueue (structure), 65
batched rebuilding, 99–101
BatchedQueue (structure), 43
binary search trees, 99
  delete, 100
  red-black, 24–28, 203
  unbalanced, 11–14, 203
BinaryRandomAccessList (structure), 123
binomial heaps, 20–24, 45–46, 70, 89–93, 198
binomial queues, *see* binomial heaps
binomial trees, 21, 118, *see also* k-nomial trees
BinomialHeap (functor), 24
Bootstrap (functor), 161
bootstrap loader, 141
BootstrappedQueue (structure), 149
bootstrapping, 141, *see also* data-structural bootstrapping
bootstrapping a compiler, 141
bottom-up mergesort, 74–78, 94–97, 202
BottomUpMergeSort (structure), 77

c-deques, *see* catenable deques
caching, 3
call-by-name, 59, 63
call-by-need, *see* lazy evaluation
call-by-value, *see* strict evaluation
catenable deques, 175–184, 191, 192
  implicit, 177–184
  signature, 176, 191
  simple, 175–178
catenable lists, 153–158, 169, 191
  signature, 153, 190
CATENABLEDEQUE (signature), 176
CatenableList (functor), 156
CATENABLELIST (signature), 153
cheap operations, 40
chef's knives, 2
complete binary leaf trees, 118
complete binary trees, 118, 132
complete cost, 60
complete k-ary leaf trees, 138
complete k-ary trees, 138
copying, 7
coroutines, 101, 102, 106, 113

credits, 40, 41

dancing bear, 2
data structure, meanings of, 3–4
data-structural bootstrapping, 169
debit inheritance, 67–68
debit passing, 174, 183
dense representations, 116, 117
DEQUE (signature), 45, 107
deques, 44, 106, 113
  banker's, 108–110, 189
  output-restricted, 107
  real-time, 111–112, 170
  signature, 45, 107, 189
destructive updates, 2
digital search trees, *see* tries
dominoes, 85
double-ended queues, *see* deques

ephemeral data structures, 2, 4, 58
execution traces, 57, 62–63
expensive operations, 40, 59, 62
ExplicitMin (functor), 23

FIFO queues, *see* queues
finite maps
  over products, 167
  over sums, 168
  signature, 163, 204
FINITEMAP (signature), 16, 163
flexible arrays, *see* random-access lists
foldr1, 155
Ford, Henry, 1
function (as opposed to operation), 4
functional programming, theoretical efficiency of, 2
functors in Standard ML, 4
future, logical, 57

garbage collection, 10
global rebuilding, 98, 101–102, 106, 113

hash table, 165, 166
HEAP (signature), 18
heap-ordered trees, 17
heaps, 169
  binomial, 20–24, 45–46, 198
  binomial, lazy, 70–71, 162
  binomial, scheduled, 89–93
  bootstrapped, 158–162, 201
  delete, 138
  leftist, 17–20, 52, 197
  leftist, weight-biased, 19
  pairing, 52–54, 56, 199
  pairing, lazy, 79–81, 200
  signature, 18, 162, 197
  skew binomial, 134–137, 162, 170, 200
  splay, 46–52, 56, 198
HEAPWITHINFO (signature), 162
higher-order functions, 76
higher-order functors, 160
hints to practitioners, 26, 44, 52, 53, 81, 89, 133, 150, 158
history, logical, 57, 61
HoodMelvilleQueue (structure), 105

imperative data structures, 2
implementation, 3
implicit recursive slowdown, 171
ImplicitCatenableDeque (functor), 181, 182
ImplicitQueue (structure), 174
incremental computations, 34, 61, 62, 67, 70
interactive systems, 83
intrinsic cost, 84

k-nomial trees, 138
knives, 2

layaway plan, 60
lazy evaluation, 2, 31, 37, 59
  syntax for, *see* $-notation
  time analysis, 60, 82
lazy numerical representations, 125–127
lazy rebuilding, 104–106
LazyBinomialHeap (functor), 71
LazyPairingHeap (functor), 80
leaf tree, 118
left-associative appends, 147
leftist heaps, 17–20, 52, 197
LeftistHeap (functor), 20
life cycle of suspensions, 61
lists, 7–10
logical future, 57, 59, 61, 85
logical history, 57, 61

memoization, 3, 37, 63
mergeable heaps, *see* heaps
mergesort, *see* bottom-up mergesort
Model T, 1
monolithic computations, 34, 61, 62, 67, 70

nested suspensions, 60, 67, 106
non-uniform recursion, 142–144, *see also* polymorphic recursion
normal-order reduction, 37
numerical representations, 115

object, 3
operation, meanings of, 4
operator, 4
or-patterns, 26
ORDERED (signature), 14

pairing heaps, 52–54, 56, 79–81, 199, 200
PairingHeap (functor), 54
parallel systems, 83
particle–antiparticle annihilation, 81
path compression, 81
path copying, 15
pattern matching, 36
  on abstract types, 180
pebble games, 97
pennants, 118, 138
persistence
  problem with amortization, 54–55
persistent data structures, 2, 7, 59, 83
persistent identity, 3
physicist's method
  limitations, 69
  traditional, 40–41, 82
  with lazy evaluation, 68–70
PhysicistsQueue (structure), 73
polymorphic recursion, 144, 170
positional number systems, 116
potential, 41
priority queues, *see* heaps

quaternary numbers, 138
QUEUE (signature), 42
queues, 97
  banker's, 64–67, 86, 106, 107, 186
  batched, 42–44, 101, 186
  bootstrapped, 146–150, 188
  Hood–Melville, 102–105, 107, 187
  implicit, 172–175, 189
  physicist's, 72–73, 104, 187
  real-time, 86–89, 106, 107
  signature, 42, 186

random-access lists, 119
  binary, 119–123, 144–147, 194, 196
  signature, 120, 194
  skew binary, 132–134, 195
RANDOMACCESSLIST (signature), 120
real-time systems, 83
realized costs, 60
realized debits, 63
RealTimeDeque (functor), 112
RealTimeQueue (structure), 88
recursive modules, 160, 161
recursive slowdown, 130, 170, 171, 184, *see also* implicit recursive slowdown
red-black trees, 24–28, 125, 203
  delete, 100
RedBlackSet (functor), 28
redundant number system, 116
reference cell, 4

ScheduledBinomialHeap (functor), 93
ScheduledBottomUpMergeSort (functor), 96
scheduling, 84–86, 106
segmented representations, 127–130
self-modification, 59
SET (signature), 12
sets
  signature, 202
shared cost, 60
sharing, 7
signatures in Standard ML, 4
SimpleCatenableDeque (functor), 178
skew binary numbers, 131–132
  canonical form, 131
skew binomial trees, 135
skew k-ary numbers, 139
SkewBinaryRandomAccessList (structure), 134
SkewBinomialHeap (functor), 137
snoc, etymology of, 42
SORTABLE (signature), 74
sortable collections, 74
  signature, 74, 202
sparse representations, 116, 117
splay trees, 46–52, 56, 59, 198
SplayHeap (functor), 50
STACK (signature), 8
steques, *see* output-restricted deques
Stream (structure), 36
STREAM (signature), 36
streams, 34–37
  signature, 36
strict evaluation, 2, 59
structural abstraction, 151–153, 170

structural decomposition, 142
structures in Standard ML, 4
suspensions, 31
  as nullary functions, 37
  life cycle of, 61
  nested, 60, 67, 106
  trivial, 35

telescoping series, 41
terminology, 3–4
Trie (functor), 165
TrieOfTrees (functor), 168
tries, 163–168, 204
trinary numbers, 138
trivial suspensions, 35

UnbalancedSet (functor), 14
uniform recursion, 142
unrealized costs, 60
unshared cost, 60

values in Standard ML, 4
version, 3
version graphs, 58
views, 180

weak updates, 100
worst-case data structures, benefits of, 83

zeroless representations, 122–125