Index

9-1-1, 2
abstract data type, see interface
adjacency list, 252
adjacency matrix, 249
algorithmic complexity attack, 132
amortized cost, 21
amortized running time, 20
ancestor, 133
array
  circular, 38
ArrayDeque, 40
ArrayQueue, 36
arrays, 29
ArrayStack, 30
asymptotic notation, 12
AVL tree, 206

B*-tree, 304
B+-tree, 304
B-tree, 286
backing array, 29
Bag, 28
BDeque, 71
Bibliography on Hashing, 128
big-Oh notation, 12
binary heap, 211
binary logarithm, 10

binary search, 272, 289
binary search tree, 140
  height balanced, 206
  partial rebuilding, 173
  random, 154
  randomized, 169
  red-black, 185
  size-balanced, 148
  versus skiplist, 105
binary search tree property, 140
binary tree, 133
  complete, 215
  heap-ordered, 212
  search, 140
binary-tree traversal, 136
BinaryHeap, 211
BinarySearchTree, 140
BinaryTree, 135
BinaryTrie, 266
binomial coefficients, 12
binomial heap, 222
black node, 190
black-height property, 190
block, 283, 284
block store, 285
BlockStore, 285
borrow, 298
bounded deque, 71
<table>
<thead>
<tr>
<th>Term</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPlusTree</td>
<td>307</td>
</tr>
<tr>
<td>breadth-first traversal</td>
<td>139</td>
</tr>
<tr>
<td>breadth-first-search</td>
<td>256</td>
</tr>
<tr>
<td>celebrity</td>
<td></td>
</tr>
<tr>
<td>ChainedHashTable</td>
<td>107</td>
</tr>
<tr>
<td>chaining</td>
<td>107</td>
</tr>
<tr>
<td>child</td>
<td>133</td>
</tr>
<tr>
<td>left</td>
<td>133</td>
</tr>
<tr>
<td>right</td>
<td>133</td>
</tr>
<tr>
<td>circular array</td>
<td>38</td>
</tr>
<tr>
<td>coin toss</td>
<td>17, 98</td>
</tr>
<tr>
<td>collision resolution</td>
<td>128</td>
</tr>
<tr>
<td>colour</td>
<td>190</td>
</tr>
<tr>
<td>Comparator</td>
<td>226</td>
</tr>
<tr>
<td>compare(a,b)</td>
<td>226</td>
</tr>
<tr>
<td>compare(x,y)</td>
<td>9</td>
</tr>
<tr>
<td>comparison tree</td>
<td>236</td>
</tr>
<tr>
<td>comparison-based sorting</td>
<td>226</td>
</tr>
<tr>
<td>complete binary tree</td>
<td>215</td>
</tr>
<tr>
<td>complexity</td>
<td></td>
</tr>
<tr>
<td>space</td>
<td>20</td>
</tr>
<tr>
<td>time</td>
<td>20</td>
</tr>
<tr>
<td>conflict graph</td>
<td>247</td>
</tr>
<tr>
<td>connected components</td>
<td>263</td>
</tr>
<tr>
<td>connected graph</td>
<td>263</td>
</tr>
<tr>
<td>contact list</td>
<td>1</td>
</tr>
<tr>
<td>conted B-tree</td>
<td>304</td>
</tr>
<tr>
<td>correctness</td>
<td>20</td>
</tr>
<tr>
<td>CountdownTree</td>
<td>183</td>
</tr>
<tr>
<td>counting-sort</td>
<td>239</td>
</tr>
<tr>
<td>credit invariant</td>
<td>302</td>
</tr>
<tr>
<td>credit scheme</td>
<td>179, 302</td>
</tr>
<tr>
<td>CubishArrayStack</td>
<td>61</td>
</tr>
<tr>
<td>cuckoo hashing</td>
<td>129</td>
</tr>
<tr>
<td>cycle</td>
<td>247</td>
</tr>
<tr>
<td>cycle detection</td>
<td>260</td>
</tr>
<tr>
<td>DaryHeap</td>
<td>223</td>
</tr>
<tr>
<td>decreaseKey(u,y)</td>
<td>222</td>
</tr>
<tr>
<td>degree</td>
<td>254</td>
</tr>
<tr>
<td>dependencies</td>
<td>22</td>
</tr>
<tr>
<td>depth</td>
<td>133</td>
</tr>
<tr>
<td>depth-first-search</td>
<td>258</td>
</tr>
<tr>
<td>deque</td>
<td>6</td>
</tr>
<tr>
<td>bounded</td>
<td>71</td>
</tr>
<tr>
<td>descendant</td>
<td>133</td>
</tr>
<tr>
<td>dictionary</td>
<td>8</td>
</tr>
<tr>
<td>directed edge</td>
<td>247</td>
</tr>
<tr>
<td>directed graph</td>
<td>247</td>
</tr>
<tr>
<td>disk access model</td>
<td>304</td>
</tr>
<tr>
<td>divide-and-conquer</td>
<td>226</td>
</tr>
<tr>
<td>DLList</td>
<td>67</td>
</tr>
<tr>
<td>doubly-linked list</td>
<td>67</td>
</tr>
<tr>
<td>Dyck word</td>
<td>28</td>
</tr>
<tr>
<td>DynamiteTree</td>
<td>183</td>
</tr>
<tr>
<td>e (Euler's constant)</td>
<td>10</td>
</tr>
<tr>
<td>edge</td>
<td>247</td>
</tr>
<tr>
<td>emergency services</td>
<td>2</td>
</tr>
<tr>
<td>Euler's constant</td>
<td>10</td>
</tr>
<tr>
<td>expected cost</td>
<td>21</td>
</tr>
<tr>
<td>expected running time</td>
<td>17, 20</td>
</tr>
<tr>
<td>expected value</td>
<td>17</td>
</tr>
<tr>
<td>exponential</td>
<td>10</td>
</tr>
<tr>
<td>Ext4</td>
<td>304</td>
</tr>
<tr>
<td>external memory</td>
<td>283</td>
</tr>
<tr>
<td>external memory hashing</td>
<td>305</td>
</tr>
<tr>
<td>external memory model</td>
<td>284</td>
</tr>
<tr>
<td>external storage</td>
<td>283</td>
</tr>
</tbody>
</table>
Index

Eytzinger’s method, 211
factorial, 11
family tree, 147
FastArrayStack, 35
Fibonacci heap, 222
FIFO queue, 5
file system, 1
finger, 103, 171
finger search
  in a skiplist, 103
  in a treap, 171
fusion tree, 281
general balanced tree, 181
git, xiv
Google, 3
graph, 247
  connected, 263
  strongly-connected, 263
$H_k$ (harmonic number), 154
hard disk, 283
harmonic number, 154
hash code, 107, 122
  for arrays, 125
  for compound objects, 123
  for primitive data, 123
  for strings, 125
hash function
  perfect, 128
hash table, 107
  cuckoo, 129
  two-level, 129
hash value, 107
hash($x$), 107
hashing
  multiplicative, 110, 129
  multiply-add, 129
  tabulation, 169
  universal, 129
  hashing with chaining, 107, 128
heap, 211
  binary, 211
  binomial, 222
  Fibonacci, 222
  leftist, 222
  pairing, 222
  skew, 222
heap order, 212
heap property, 159
heap-ordered binary tree, 212
heap-sort, 233
height
  in a tree, 133
  of a skiplist, 87
  of a tree, 133
height-balanced, 206
HFS+, 304
I/O model, 304
in-order number, 148
in-order traversal, 148
in-place algorithm, 243
incidence matrix, 262
indicator random variable, 17
interface, 4
Java Collections Framework, 26
Java Runtime Environment, 60
leaf, 133
left child, 133
left rotation, 161
Index

left-leaning property, 194
left-leaning red-black tree, 194
leftist heap, 222
LIFO queue, 5, see also stack
linear probing, 114
LinearHashTable, 114
linearity of expectation, 17
linked list, 63
doubly-, 67
singly-, 63
space-efficient, 71
unrolled, see also SEList
List, 6
logarithm, 10
binary, 10
natural, 10
lower-bound, 235
map, 8
matched string, 28
MeldableHeap, 217
memcpy(d, s, n), 36
memory manager, 60
merge, 187, 299
merge-sort, 84, 226
min-wise independence, 169
MinDeque, 85
MinQueue, 85
MinStack, 85
modular arithmetic, 37
multiplicative hashing, 110, 129
multiply-add hashing, 129
n, 22
natural logarithm, 10
no-red-edge property, 190
NTFS, 304
number
in-order, 148
post-order, 148
pre-order, 148
O notation, 12
open addressing, 114, 128
Open Source, xiii
ordered tree, 133
pair, 8
pairing heap, 222
palindrome, 83
parent, 133
partial rebuilding, 173
path, 247
pedigree family tree, 147, 222
perfect hash function, 128
perfect hashing, 128
permutation, 11
random, 154
pivot element, 230
planarity testing, 262
post-order number, 148
post-order traversal, 148
potential, 48
potential method, 48, 80, 205
pre-order number, 148
pre-order traversal, 148
prime field, 126
priority queue, 5, see also heap
probability, 15
queue
FIFO, 5
LIFO, 5
Index

priority, 5
quicksort, 230
radix-sort, 241
RAM, 18
random binary search tree, 154
random permutation, 154
randomization, 15
randomized algorithm, 15
randomized binary search tree, 169
randomized data structure, 15
RandomQueue, 60
reachable vertex, 247
recursive algorithm, 136
red node, 190
red-black tree, 185, 194
RedBlackTree, 194
remix, xiii
right child, 133
right rotation, 161
rooted tree, 133
RootishArrayStack, 49
rotation, 161
run, 118
running time, 20
  amortized, 20
  expected, 17, 20
  worst-case, 20
scapegoat, 173
ScapegoatTree, 174
search path
  in a BinaryTrie, 266
  in a binary search tree, 140
  in a skiplist, 88
secondary structure, 275
SEList, 71
sentinel node, 88
Sequence, 184
share, xiii
simple path/cycle, 247
singly-linked list, 63
size-balanced, 148
skew heap, 222
skiplist, 87
  versus binary search tree, 105
SkipList, 93
SkipListSSet, 90
SLList, 63
social network, 1
solid-state drive, 283
sorting algorithm
  comparison-based, 226
sorting lower-bound, 235
source, 247
space complexity, 20
spanning forest, 263
speciation event, 147
species tree, 147
split, 187, 290
square roots, 56
SSet, 9
stable sorting algorithm, 241
stack, 5
std::copy(a0,a1,b), 36
Stirling's Approximation, 11
stratified tree, 280
string
  matched, 28
strongly-connected graph, 263
successor search, 9
System.arraycopy(s,i,d,j,n), 36
Index

tabulation hashing, 121, 169

XFastTrie, 272

XOR-list, 82

target, 247

YFastTrie, 275
tiered-vector, 59
time complexity, 20

traversal
  breadth-first, 139
  in-order, 148
  of a binary tree, 136
  post-order, 148
  pre-order, 148

Treap, 159

TreapList, 172
tree, 133
  d-ary, 222
  binary, 133
  ordered, 133
  rooted, 133

tree traversal, 136

Treque, 60

two-level hash table, 129

underflow, 295

universal hashing, 129

universal sink, 263

unrolled linked list, see also SEList

USet, 8

van Emde Boas tree, 280

vertex, 247

wasted space, 54

web search, 1

WeightBalancedTree, 183

word, 19

word-RAM, 18

worst-case running time, 20