INDEX

**Numbers**

32-bit applications, 297–298
64-bit applications, 297–298

**Symbols**

<< (left arrow), shift operators, 226
>> (right arrow), shift operators, 226

**A**

abstract classes, 184–186
accessibility property, distributed databases, 211
accomplishments, showing you can get things done, 6
ACID (atomicity, consistency, isolation, and durability) properties, 210–211
actions (or capabilities or methods), class. See methods
acyclic lists, distinguishing cyclic lists from, 70–73
adjacency list, representation of graph data, 82
adjacency matrix, representation of graph data, 82
aggregates, SQL statements, 208
AI (artificial intelligence), 244–245
algorithms
determining best sorting algorithm, 150–153
finding best algorithm in Big-O analysis, 38–39
for finding mth-to-last element in linked list, 62–64
insertion sorts, 145–146
list-flattening and, 66–67
list-unflattening and, 69–70
memory usage of, 39
merge sorts, 148–149
optimization using, 37–38
overview, 143–144
quicksort algorithm, 146–148
selection sorts, 144–145
analyzing your solution
coding skills and, 29
interactive nature of problem-solving in, 31
language requirements and, 30–31
memory footprint analysis, 39–40
problem requirements and, 30
problem-solving steps, 32–33
runtime analysis (Big-O), 35–39
scenario for, 29–30
summary, 40
what to do if you get stuck, 34
ancestors
finding lowest common ancestor in tree structure, 87–88
of trees, 77
AND operator
bitwise operators, 225
determining number of 1 bits in binary representation of an integer, 235–237
APIs (application programming interfaces), 185
application developer, know your aptitudes and likes, 1
application programming interfaces (APIs), 185
argument passing, in C++, 292–293
arms-length recursion, 133
Array object, JavaScript, 102
arrays
  in C#, 102
  in C and C++, 100–101
dynamic arrays for stack implementation, 49
finding the first nonrepeated character, 106–109
insertion sort and, 145
in Java, 101
in JavaScript, 102, 105
overview, 99–100
removing specified characters from strings, 109–111
selection sort and, 144
summary, 124
using array sorting routine to convert tree into a heap, 88–91
UTF-8, 121–123
artificial intelligence (AI), 244–245
ASCII
  removing specific characters from ASCII string, 109–111
  as subset of UTF-8, 103
assumptions
  logic and, 260–261
  problem-solving and, 267–269
atomicity property, ACID properties in database transactions, 210
attributes, class, 181
attributes (columns), relational database tables, 203
augmentation, 250
AVF, SQL aggregates, 208
AVL trees, 92

base classes, inheritance and, 183
Basic Multilingual Plane (BMP), Unicode and, 108
behavioral design patterns
  Iterator pattern, 197
  Observer pattern, 197
BFS (breadth-first search)
  overview of, 80–81
in “Six Degrees of Kevin Bacon” game, 95
big data, 239
big-endian, determining endianness of computer, 233–234
Big-O analysis
  applying, 35–36
  how it works, 36–37
  optimization using, 37–38
  procedure for, 38
  selecting best performing algorithm, 38–39
binary numbers
  binary two’s complement notation for, 224–225
determining number of 1 bits in binary representation of an integer, 235–237
binary search
  applying to heavy marble problem, 271
  recursion applied to, 129–131
binary trees. See also BSTs (binary search trees)
  converting to heap, 88–91
  overview, 77
  unbalanced, 91–92
binary two’s complement notation, for numbers, 224–225
binomial distribution, 241
bit manipulation
  binary two’s complement notation and, 224–225
  bitwise operators, 225–226
determining if computer is big-endian or little-endian, 233–234
determining number of 1 bits in binary representation of an integer, 235–237
optimizing using shift operators, 226
bitwise operators, 225–226
blacklisting, Web application security and, 299
blogs, developing marketable skills, 8
BMP (Basic Multilingual Plane), Unicode and, 108
boat and pier problem, graphical/spatial brainteasers, 276–278
Bonferroni correction, 248
bonuses, referrals and, 10
bookstores, researching job market, 4
brainteasers
being wary of simple problems, 262
bridge crossing problem, 266–269
estimation problems, 262–263
gas stations in U.S. problem, 273–274
heavy marble problem, 269–273
not being intimidated by, 261–262
open lockers problem, 263–265
overview of, 259
summary, 274
tackling, 259–263
three light switch problem, 265–266
brainteasers, graphical or spatial
boat and pier problem, 276–278
burning fuse problem, 283–285
cube counting problem, 278–282
drawing as first step in solving, 275–276
escaping oncoming train problem, 286–287
fox and duck problem, 282–283
overview, 275
summary, 287
breadth-first search (BFS)
overview of, 80–81
in “Six Degrees of Kevin Bacon” game, 95
bridge crossing problem, brainteasers, 266–269
BSTs (binary search trees)
finding lowest common ancestor in tree structure, 87–88
vs. hash tables, 301–302
overview, 78–80
reorganizing an unbalanced binary search tree, 91–92
bug-finding problem, linked lists, 60–62
Builder pattern, 195–196
burning fuse problem, graphical/spatial brainteasers, 283–285
busy waiting problem in concurrency, 170–172
CPU time wasted by, 174
C language
arrays in, 100–101
inserting/deleting elements in linked lists, 46–48
linked list problems and, 41
memory allocation in, 22
pointer misuse in, 45
strings in, 103–104
writing C function to determine endianness of computer, 233–234
C# language
applying languages in programming interviews, 30
arrays in, 102
converting integers to strings, 116
garbage collection and, 296
multiple inheritance disallowed, 188
not supporting friend classes, 292
OOP (object-oriented programming) and, 181
strings in, 105
C++ language
applying languages in programming interviews, 30–31
argument passing, 292–293
arrays in, 100–101
comparing macros and inline functions, 294–295
comparing with Java, 291
destructor methods in, 182
example of use of virtual methods, 186–188
implementing singly linked list, 42–43
C++ language (continued)
inserting/deleting elements in linked lists, 48
linked list problems and, 41
multiple inheritance in, 188
OOP (object-oriented programming) and, 181
pointer misuse in, 45–46
strings in, 104
C99 language, 294–295
capabilities, class. See methods
career-related questions
“What Are Your Career Goals?,” 306
“What Can You Tell Me About Your Experience?,” 306
Cartesian coordinate system
raster pixel displays and, 223
scan conversion and, 227–228
Cartesian product, 207
char object, in C and C++, 103
characters
converting numeric values to, 118
converting to numeric values, 116–118
finding the first nonrepeated character, 106–109
removing specific characters from ASCII string, 109–111
reversing order of words in string, 112–115
sequence of. See strings
child lists
list-flattening and, 66–67
list-unflattening and, 68–69
child nodes, of binary trees, 77
circles, exercise drawing upper-eighth of, 227–229
circular linked lists, 44
clarity, in résumé, 319
classes
abstract classes, 184–186
attributes and actions, 181–182
creational design patterns and, 195–196
definition in object-oriented languages, 75–76
friend classes, 292
inheritance and polymorphism and, 183–184
cleartext passwords, 299
coding
explaining your approach and testing, 33
know your aptitudes and likes, 2
programming interviews covering coding skills, 29–30
columns (attributes), relational database tables, 203
combinations, of strings, 134–136
combinatorial mathematics, 138
common key, 206
communication, being interactive in
programming interviews, 31
companies
contacting for job application, 11–12
finding for job application, 9–10
know your aptitudes and likes in selecting, 2–3
“What Why Do You Want to Work for This Company?,” 311
company database, 212–214
comparator function, using with sorting routines, 156–158
CompareToAll function, in Big-O analysis, 35–37
CompareToMax function, in Big-O analysis, 35–37
compensation. See also salary
knowing what you want in compensation package, 307–308
negotiating compensation package, 16
compilers, handling inline functions, 294
conciseness, in résumé, 319
concurrency
ATM example, 168–170
busy waiting problem, 170–172
deadlocks, 167
dining philosophers problem, 175–178
monitors and semaphores in thread synchronization, 166–167
overview, 165
producer/consumer threads and, 172–175
summary, 179
system threads vs, user thread, 166
threads and, 165–166
certainty intervals, 242
consistency property, ACID properties in
database transactions, 210
consistency property, distributed databases,
211
constructor methods, for creating objects,
182–183
contacts
getting referrals from, 10
maintaining even while rejecting offers, 17
context switches, in thread management, 166
cooperative model, for thread management,
166
copy constructor, 293
COUNT, SQL aggregates, 208
counting semaphores, locking shared
resources, 167
co-workers, getting referrals from, 10
createStack function, stack implementation
in linked lists, 51–54
creational design patterns
Builder pattern, 195–196
overview, 194
Singleton pattern, 195
credentials, developing marketable skills, 5
cryptography
  cryptographic hashes, 300
  public and shared key approaches, 301
cube counting problem, graphical/spatial
brainteasers, 278–282
curriculum vitae (CV), 317
cyclic lists, distinguishing from acyclic lists,
70–73
list-flattening and, 66
list-unflattening and, 68
trees. See trees
using in problem-solving, 34
database management systems (DBMS), 204
  databases
  company and employee, 212–214
  distributed databases, 211–212
  exercise writing SQL statement that returns
  nonapartment addresses, 216–218
  hybrid key-value/column databases,
  209–210
NoSQL database, 208–209
object databases, 209
overview of, 203
relational, 203–204
SQL databases, 204–208
transactions in, 210–211
DBMS (database management systems), 204
deadlocks
  in dining philosopher problem, 178
  in thread management, 167
debugging skills, know your aptitudes and
  likes, 2
Decorator pattern
  comparing with inheritance, 201
  as Wrapper pattern, 197
deep learning, 244–245
defered initialization, Singleton pattern and,
  199–200
deleteStack function, stack implementation
  in linked lists, 51–54
depth-first search (DFS)
  overview of, 81
  in “Six Degrees of Kevin Bacon” game, 94
descendants, of trees, 77
descriptive statistics, 241–242
design patterns
  applying Singleton pattern to problem,
  199–200
  comparing use of Decorator pattern with
  inheritance, 201
creational patterns, 195–196
efficient updating of Observer pattern, 202
data science, 239–240
data scientist, 240
data structures
graphs. See graphs
design patterns (continued)
overview, 193
reasons for using, 193–194
summary, 202

Design Patterns: Elements of Reusable Object Oriented Software (Gamma, Helm, Johnson, and Vlissades), 193, 194
derestructor methods, objects, 182–183
DFS (depth-first search)
overview of, 81
in “Six Degrees of Kevin Bacon” game, 94
digest strings, Web application security and, 300
Dijkstra’s algorithm, 95
dining philosophers problem, 175–178
directed graphs, 82
display, raster pixel displays and, 223
distributed databases, 211–212
distribution, 241
divide-and-conquer algorithms
merge sort as, 148
quicksort as, 146
doctorates, developing marketable skills, 5
domain-specific knowledge, outsourcing risk and, 5
doubly linked lists
flattening, 65–68
overview of, 44
unflattening, 68–70
drawing, as approach to solving graphical/spatial brainteasers, 275–276
dressing, for interviews, 14
duplicates, removing, 25–26
durability property, ACID properties in database transactions, 210
dynamic arrays
overview. See arrays for stack implementation, 49
e-mail, submitting résumé to companies via, 11–12
employee database, 212–214
employees, getting referrals from, 10
employers, getting referrals from, 10
encapsulation, 182
endianness, writing C function to determine endianness of computer, 233–234
error handling
finding/fixing bugs in removeHead, 60–62
stack implementation in linked lists and, 50–51
estimation problems
overview, 262–263
problem determining number of gas stations in U.S., 273–274
event semaphores, locking shared resources, 167
event threads, 166
eventual consistency, 211
event-related questions, in interviews, 304

F
factorial, implementing via recursion, 126–127
feature engineering, 245
finalizer methods, objects, 183
FizzBuzz, 23–24
floating-point values, rounding and, 224
foreign key constraints, in relational databases, 204
Fortran language, 31
forums, developing marketable skills, 7
fox and duck problem, graphical/spatial brainteasers, 282–283
friend classes, 292
friend keyword, applying to functions or classes, 292
FROM clause, SQL statements, 207
functions
in Big-O analysis, 35–37
comparator function, 156–158
copying strings in C, 103
inline functions, 294–295
recursive function, 125
stack implementation in linked lists and, 51–54

G

The Gang of Four, 193–194
garbage collection, 296–297
gas stations in U.S. problem, brainteasers, 273–274
Gaussian distribution, 241
GitHub, managing online profile, 7
Go language, 31
goals, “What Are Your Career Goals?,” 306
graduate degrees, developing marketable skills, 5
graphical/spatial brainteasers
  boat and pier problem, 276–278
  burning fuse problem, 283–285
  cube counting problem, 278–282
drawing as first step in solving, 275–276
drawing as first step in solving, 275–276
escaping oncoming train problem, 286–287
fox and duck problem, 282–283
overview, 275
summary, 287
graphics
  exercise drawing upper-eighth of a circle, 227–229
  exercising determining if two rectangles overlap, 229–232
  overview of, 223–224
graphs
  BFS (breadth-first search) in, 95
  overview, 75
  in “Six Degrees of Kevin Bacon” game, 94–95
types of, 82
green threads, 166

H

hash tables
  vs. BSTs, 301–302
finding the first nonrepeated character, 106–109
removing specific characters from strings, 109–111
hashes, Web application security and, 300
HCI (human computer interaction), know your aptitudes and likes, 1–2
head element
  in doubly linked lists, 44
finding/fixing bugs in removeHead, 60–62
null vs. cyclic lists and, 70–73
in singly linked lists, 42
tracking in singly linked list, 44–46
headhunters, working with, 10–11
heaps
  converting binary tree to, 88–91
  overview, 80
heavy marble problem, brainteasers, 269–273
high-performance computing (HPC), 172
Horner’s Rule, 117–118
HPC (high-performance computing), 172
human computer interaction (HCI), know your aptitudes and likes, 1–2
hybrid key-value/column databases, 209–210
hybrid merge sorts, 149
hyperparameters, 245

I

inferential statistics, 241–242
inheritance
  classes and, 183–184
  comparing Decorator pattern with, 201
  multiple inheritance, 188–189
  passing methods and, 295–296
inline functions, comparing macros with, 294–295
inner join, SQL tables, 207
inorder traversal
  defined, 81
  process of, 85
in-place sorting
  overview, 144
  selection sort and, 145–146
INSERT statement, 205, 212
**insertAfter**, maintaining tail pointer in linked list, 57–58

inserting/deleting elements, into/from linked lists, 46–48

insertion sorts, 145–146

integers
  - converting to strings, 118–121
  - determining number of 1 bits in binary representation of an integer, 235–237
  - rounding, 224

interactivity
  - in problem-solving, 32
  - in programming interviews, 31

**interface** keyword, Java, 185

interfaces
  - *vs.* abstract classes, 184–186
  - know your aptitudes and likes, 1–2

Internet, submitting résumé to companies via, 11

internships, developing marketable skills, 6

interviews
  - dressing for, 14
  - on-site interviews, 13
  - screening interviews, 12–13

is-a relationships, abstract classes and, 186

isolation property, ACID properties in database transactions, 210

iteration, compared with recursive solutions, 128

**Iterator** pattern, 197

### J

jagged arrays, in C#, 102

Java language
  - applying languages in programming interviews, 30
  - applying recursion to telephone word problem, 140–142
  - arrays in, 101
  - comparing with C++, 291
  - converting integers to strings, 116
  - creating monitor for thread management, 169–170
  - defining interfaces in, 185
  - garbage collection and, 296
  - implementing singly linked list, 43
  - multiple inheritance disallowed, 188
  - not supporting friend classes, 292
  - OOP (object-oriented programming) and, 181
  - permutation of strings, 133
  - strings in, 104–105
  - thread-safe and nthread-safe classes on, 167
  - tracking head element of singly linked list, 45

**JavaScript**
  - applying languages in programming interviews, 30
  - arrays in, 102, 105
  - use of OOP in, 181

job application
  - accepting/rejecting job offers, 17
  - contacting companies, 11–12
  - dressing for interviews, 14
  - finding companies, 9–10
  - getting referrals, 10
  - job fairs, 12
  - negotiating salary, 15–16
  - on-site interviews, 13
  - overview of, 9
  - preparing for. See preparation, for job search
  - recruiter’s role in, 14–15
  - screening interviews and, 12–13
  - summary, 17
  - technology-driven sites, 12
  - “Why Should We Hire You?,” 310
  - working with headhunters, 10–11

job fairs, learning about potential employers, 12

job offers
  - accepting/rejecting, 17
  - negotiating salary, 15–16

joins, SQL tables, 207

### K

kernel-level threads, 166

keys
  - in relational databases, 203–204
  - for sort order, 144
know yourself, 1–3
knowledge-based questions
32-bit vs. 64-bit applications, 297–298
argument passing, 292–293
C++ vs. Java, 291
cryptography, 301
friend classes, 292
garbage collection, 296–297
hash tables vs. BSTs, 301–302
inheritance, 295–296
macros and inline functions, 294–295
MapReduce, 302
network performance, 298
preparing for, 289–290
summary, 302
types of problems, 290–291
Web application security, 298–300

languages. See programming languages
last-in-first-out (LIFO), stacks and, 49
latency, network performance and, 298
lazy initialization (lazy loading), Singleton
learning, developing marketable skills, 6
least-significant bit (LSB), in determining
endianness of computer, 233–234
leaves, of trees, 77
left node, child nodes of binary trees, 77
LIFO (last-in-first-out), stacks and, 49
linked lists
bug-finding in, 60–62
distinguishing cyclic from acyclic lists, 70–73
finding mth-to-last element in, 62–64
flattening, 65–68
inserting/deleting elements, 46–48
maintenance of tail pointer, 54–60
overview, 41
stack implementation, 48–54
summary, 73
tracking head element of singly linked list, 44–46
traversing a list, 46
types of, 42–44
unflattening, 68–70
LinkedIn, managing online profile, 7
Lisp language
applying languages in programming
interviews, 31
garbage collection and, 296
lists, linked. See linked lists
little-endian, determining endianness of
computer, 233–234
livelocks, in dining philosopher problem, 177
logic, assumptions and, 260–261
logical operators, compared with bitwise
operators, 225
lookups
finding the first nonrepeated character, 106–109
using binary search trees, 79–80
LSB (least-significant bit), in determining
endianness of computer, 233–234

machine learning, 244–245
macros, comparing inline functions with, 294–295
management
know your aptitudes and likes, 2
outsourcing risk and, 5
managers, résumé for, 323–332
map phase, MapReduce, 302
MapReduce, 302
mark and sweep implementation, of tracing
  garbage collector, 297
market, job
developing marketable skills, 5–6
getting information regarding, 3–4
outsourcing risk and, 4–5
master’s degrees, developing marketable
  skills, 5
mathematics, combinatorial, 138
max heap, 80
MAX statement, 215–216
MD5 cryptographic hash, 300
memory allocation, in C, 22
memory footprint analysis, 39–40
merge sort
determining best sorting algorithm, 150–153
overview, 148–149
methods
class, 181
constructors/destructors/finalizers, 182–183
finalizer methods, 183
inclusion and, 296
interfaces vs. abstract classes, 184–186
polymorphism and, 183
virtual methods, 186–188
min heap, 80
mobile programming, 23
model-view controller (MVC), separation of responsibilities, 197
monitors
busy waiting problem and, 171
creating in Java, 169–170
thread synchronization and, 166–167
Monte Carlo methods, 254–258
most-significant bit (MSB), in determining endianness of computer, 233–234
move constructor, 293
MSB (most-significant bit), in determining endianness of computer, 233–234
mth-to-last element, linked list problems, 62–64
multidimensional arrays, in Java, 101
multi-key sort problem, 155–156
multiple inheritance, 188–189
multiple testing problem, 248
multithreaded applications, 170–172
mutex (mutual exclusion) semaphore, 167
mutual exclusion (mutex) semaphore, 167
MVC (model-view controller), separation of responsibilities, 197

N
native threads, 166
negotiation, of salary, 15–16
nested parentheses, 26–27
network performance, 298
neural network, 244
nodes

O
object databases, 209
object-oriented programming. See OOP
(objects-oriented programming)
objects
as class instance, 181–182
of binary search trees, 79–80
of binary trees, 77
determining height of trees and, 83–84
in “Six Degrees of Kevin Bacon” game, 93–96
of trees, 75–76
nonparametric statistics, 243
nontechnical people, phone screens by, 20–21
non-technical questions
“Do You Have Any Questions for Me?,” 311
overview, 303
reasons for, 303–304
“What Are Your Career Goals?,” 306
“What Can You Tell Me About Your Experience?,” 306
“What Do You Want to Do?,” 304–305
“What Is Your Favorite Programming Language?,” 305–306
“What Is Your Salary History?,” 310
“What Is Your Work Style?,” 306
“What Salary Are You Expecting?,” 307–310
Why Are You Looking to Change Jobs?,” 306–307
Why Do You Want to Work for This Company?,” 311
Why Should We Hire You?,” 310
normal distribution, 241
NoSQL database, 208–209
NOT operator, bitwise operators, 225

NULL
cyclic lists and, 70–73
maintaining tail pointer in linked list, 54–57
numeric values
binary two’s complement notation of, 224–225
converting characters to, 116–118
converting to characters, 118

of binary search trees, 79–80
of binary trees, 77
determining height of trees and, 83–84
in “Six Degrees of Kevin Bacon” game, 93–96
of trees, 75–76
nonparametric statistics, 243
nontechnical people, phone screens by, 20–21
non-technical questions
“Do You Have Any Questions for Me?,” 311
overview, 303
reasons for, 303–304
“What Are Your Career Goals?,” 306
“What Can You Tell Me About Your Experience?,” 306
“What Do You Want to Do?,” 304–305
“What Is Your Favorite Programming Language?,” 305–306
“What Is Your Salary History?,” 310
“What Is Your Work Style?,” 306
“What Salary Are You Expecting?,” 307–310
Why Are You Looking to Change Jobs?,” 306–307
Why Do You Want to Work for This Company?,” 311
Why Should We Hire You?,” 310
normal distribution, 241
NoSQL database, 208–209
NOT operator, bitwise operators, 225

NULL
cyclic lists and, 70–73
maintaining tail pointer in linked list, 54–57
numeric values
binary two’s complement notation of, 224–225
converting characters to, 116–118
converting to characters, 118

object databases, 209
object-oriented programming. See OOP
(objects-oriented programming)
objects
as class instance, 181–182
constructors/destructors, 182–183
creational design patterns and, 195–196
passing methods through inheritance, 296
Observer pattern
efficient updating of, 202
overview, 197
offers. See job offers
offshoring, avoiding loss of job due to, 4–5
O(n) algorithm
finding the first nonrepeated character, 107
removing specified characters from ASCII string, 110–111
online algorithm, processing data as it becomes available, 151–152
online job sites, getting market information from, 3
online profile, managing, 7–8
on-site interviews, 13
OOP (object-oriented programming)
class definitions, 75–76
classes and objects, 181–182
constructors/destructors, 182–183
inheritance and polymorphism, 183–184
interfaces vs. abstract classes, 184–186
multiple inheritance, 188–189
overview, 181
resource management, 189–191
summary, 191
virtual methods, 186–188
open lockers problem, brainteasers, 263–265
open source
know your aptitudes and likes, 3
working on open source project to develop marketable skills, 5
operating systems, native threads and, 166
OR operator, bitwise operators, 225
outer join, SQL tables, 207
outsourcing, avoiding loss of job due to, 4–5
overfitting, 249
overrides, polymorphism and, 183
parent node, of trees, 77
partitionability property, distributed databases, 211
passwords, Web application security and, 300
performance
memory footprint analysis, 39–40
runtime analysis (Big-O), 35–39
Perl language, 31
permutations, of strings, 131–134
phone interviews. See screening interviews
phone screens
FizzBuzz, 23–24
memory allocation in C, 22
mobile programming, 23
nested parentheses, 26–27
by nontechnical people, 20–21
overview, 19
problems with, 22–27
recursion trade-offs, 22–23
removing duplicates, 25–26
reversing strings, 24–25
by software engineers, 19–20
summary, 27
taking, 21–22
PHP language, 31
Pi, calculating, 254–258
pivot value, in quicksorts, 146–148, 158–161
pixels, raster pixel displays and, 223
pointers
argument passing and, 293
for finding mth-to-last element in linked list, 64
inserting/deleting elements in linked lists, 46–48
maintaining tail pointer in linked list, 54–60
passing to stacks, 49
pointer constants, 100
to tail of list, 67–68
tracking head element of singly linked list, 44–46
to tree nodes, 75
polymorphism
overview, 183–184
virtual methods used for, 188
pop, removing elements from stacks, 49–54

pancake sort problem, 161–163
parametric statistics, 243
parent classes, 183
population statistic, 241–242
postorder traversal, 81, 85
preemptive threading, 166
preorder traversal
  defined, 81
  process of, 84–85
  without using recursion, 85–87
preparation, for job interview, 311
preparation, for job search
  developing marketable skills, 5–6
  know the market, 3–4
  know yourself, 1–3
  managing online profile, 7–8
  outsourcing risk and, 4–5
  showcasing accomplishments, 6
  summary, 8
preparation, for knowledge-based questions,
  289–290
The Preparation of Programs for an
  Electronic Digital Computer, 237
prepared statements, SQL, 299
primary key, in relational databases, 203–204
probability, 240–241
problem-solving
  analyzing your solution, 34–35
  interactive nature of, 31
  programming interviews and, 30
  runtime analysis (Big-O), 35–39
  steps, 32–33
  what to do if you get stuck, 34
procedure, for Big-O analysis, 38
producer/consumer threads, example of
  concurrency problem, 172–175
professional development, courses in, 4
programming, mobile, 23
programming interviews, analyzing your
  solution, 34–35
programming languages
  programming interviews and, 30–31
  “What Is Your Favorite Programming
    Language?,” 305–306
  what to do if you get stuck, 34
proofreading résumé, 321
properties (or states), class, 181
pseudorandom number generator, 246
public key cryptography, 301
Publish-Subscribe pattern, 197
push, adding elements to stacks, 49–54
p-value, 242–243
Python language
  applying languages in programming
    interviews, 30
  garbage collection and, 296
quality assurance (QA), know your aptitudes
  and likes, 2
queries
  exercise writing SQL statement that returns
    nonapartment addresses, 216–218
  in relational databases, 203
questions
  knowledge-based. See knowledge-based
    questions
  non-technical. See non-technical questions
  offering specific and thorough responses to,
    290
  preparing for, 289–290
quicksort
  determining best sorting algorithm, 150–153
  optimizing, 158–161
  overview, 146–148
race conditions, in dining philosopher
  problem, 177
RAII (Resource Acquisition Is Initialization),
  190
rainbow table, Web application security and,
  300
random number generators
  calculate Pi, 254–258
  irreproducible results problem, 247–248
  overview, 245–246
  roll the dice problem, 251–254
  study more; know less problem, 249–251
  summary, 258
raster pixel displays
  Cartesian coordinate system and, 223
  scan conversion, 227–228
recruiters
  dealing with high-pressure tactics of, 15
  maintaining contact with even while rejecting offers, 17
  negotiating compensation package, 16
  role of, 14–15
  working with headhunters, 10–11
rectangles, exercising determining overlap, 229–232
recursion
  applying to binary search, 129–131
  applying to telephone word problem, 137–142
  arms-length recursion, 133
  combinations of a string, 134–136
  implementing depth-first traversal, 81
  iterative solutions compared with, 128
  merge sort, 148–149
  overview, 125–129
  permutations of a string, 131–134
  preorder traversal without using, 85–87
  selection sort and, 144–145
  summary, 142
  thinking recursively when using binary search trees, 79–80
  trade-offs for, 22–23
  recursive case, 125–126
  recursive function, 125
red-black trees, 92
reduce phase, MapReduce, 302
reference counting, garbage collection method, 296
references
  argument passing and, 293
  inserting/deleting elements in linked lists, 46–48
  tracking head element of singly linked list, 44–46
referential integrity, in relational databases, 204
referrals, getting for job application, 10
regression, 244
relational databases
  exercise writing SQL statement that returns nonapartment addresses, 216–218
  overview, 203–204
  SQL databases, 204–208
  transactions in, 210–211
  removeHead, finding/fixing bugs in, 60–62
  representation learning, 245
  representative sample, 242
  Resource Acquisition Is Initialization (RAII), 190
resource management, OOP, 189–191
résumé
  clarity and conciseness in, 319
  examples, 314–332
  financial incentives for submitting, 10
  keeping it relevant, 320
  keeping it short, 317–318
  listing the right information in, 318
  for managers and senior developers, 323–332
  proofreading, 321
  reversing chronological order in, 321
  selling yourself, 317
  submitting to companies, 11–12
  tailoring to the position, 329
  technical, 313–323
  reversing strings, 24–25
  right node, child nodes of binary trees, 77
  right rotation, balancing binary search trees, 92
rollback, of database transactions, 210
root node, of trees, 76
rounding off integers, 224
rows (tuples), relational database tables, 203
Ruby language, 31
Ruby on Rails, 203
runtime analysis (Big-O). See Big-O analysis
r-value reference, 293
S
salary
  negotiating, 15–16
  recruiter’s role in, 15
salary (continued)
“What Is Your Salary History?,” 310
“What Salary Are You Expecting?,” 307–310
salt, cryptographic hash, 300
Scala language, 31
scan conversion, converting geometric drawing to pixel-based raster image, 227
schema
relational database table definitions, 203
in SQL database, 218–222
screening interviews, 12–13
search algorithms
BFS (breadth-first search), 80–81
DFS (depth-first search), 81
search engine optimization (SEO), managing online profile, 7
searches
applying recursion to binary search, 129–131
binary search trees. See BSTs (binary search trees)
security, Web application security, 299
SELECT statement, SQL statements, 205–206
selection sort
determining best sorting algorithm, 150–153
implementing a stable version, 153–155
overview, 144–145
selling yourself, in your résumé, 317
semaphores
busy waiting problem and, 171
thread synchronization and, 166–167
senior developers, résumé for, 323–332
SEO (search engine optimization), managing online profile, 7
SHA cryptographic hashes, 300
shared key (symmetric) cryptography, 301
sharing resources, 167
shift operators, bitwise operators, 226
shuffle phase, MapReduce, 302
sign extension, 226
simplification, as problem-solving technique, 262
Simula language, in history of OOP, 181
Singleton pattern
applying, 199–200
deferred initialization, 199–200
implementing, 198–200
overview of, 195
singly linked lists
for finding mth-to-last element in, 62–64
finding/fixing bugs in, 60–62
overview of, 42–43
tracking head element of, 44–46
“Six Degrees of Kevin Bacon” game, 93–96
skill set
developing marketable skills, 5–6
listing the right information in résumé, 318
reversing chronological order in résumé, 321
tailoring résumé to the position, 329
Smalltalk language, in history of OOP, 181
social networks
job market and, 3
managing online profile, 7
software architecture, know your aptitudes and likes, 2
software development firms, outsourcing risk and, 4
software engineers, phone screens by, 19–20
sorting
algorithms for, 143–144
determining best sorting algorithm, 150–153
implementing a stable version of selection sort algorithm, 153–155
insertion sorts, 145–146
merge sorts, 148–149
multi-key sort problem, 155–156
optimizing quicksort algorithm, 158–161
overview, 143
pancake sort problem, 161–163
quicksort algorithm, 146–148
selection sorts, 144–145
stabilizing a sort routine, 156–158
stack implementation in linked lists, 49–54
summary, 163
spatial brainteasers. See graphical/spatial brainteasers
spinlocks, 172
SQL (Structured Query Language)
exercise writing SQL statement that returns nonapartment addresses, 216–218
overview, 204–208, 212
prepared statements, 299
SQL injection attacks, 299
stability
implementing a stable version of selection sort algorithm, 153–155
making a sort routine stable, 156–158
of sorting algorithms, 144
Stack Overflow, managing online profile, 7
stacks
linked list problems, 49–54
using with recursive algorithms, 128
standard deviation, 241
standard library, sorting routine applied to, 155
Standard Template Library, C++ language, 104
standard uniform distribution, 246
states (properties), class, 181
static arrays, compared with dynamic, 100
statistical power, 243
statistics
confidence intervals, 242
descriptive statistics, 241–242
inferential statistics, 241–242
nonparametric statistics, 243
overview, 240–241
parametric statistics, 243
summary, 258
tests, 242–243

String class
C++ language, 104
Java and C# languages, 104–105
strings
in C#, 105
in C and C++, 103–104
combinations of, 134–136
converting integers to, 118–121
converting to integers, 116–118
finding the first nonrepeated character, 106–109
in Java, 104–105
overview, 102–103
permutations of, 131–134
removing specific characters from ASCII string, 109–111
reversing order of words in, 24–25, 112–115
summary, 124
UTF-8, 121–123
strlcpy function, copying strings in C, 103
struct, in singly linked lists, 42
Structured Query Language. See SQL
Student's t-test, 243
subclasses, inheritance and, 183
SUM, SQL aggregates, 208
supervised machine learning, 244
symmetric key cryptography (shared key), 301
synchronized keyword, creating monitor for thread management in Java, 169–170
system threads, 166
System.array base calls, in C#, 102
systems programmer, know your aptitudes and likes, 1

tables, in relational databases, 203
tail call elimination, 126
tail element
in doubly linked lists, 44
maintaining tail pointer in linked list, 54–60
pointer to, 67–68
in singly linked lists, 42
tail-recursive functions, 126
technical résumé, 313–323
technology-driven sites, 12
telephone word problem, 137–142
ternary logic, in SQL, 216–218
test set, 245
testing, know your aptitudes and likes, 2
thread synchronization
ATM example, 168–170
monitors and semaphores in, 166–167
threads
ATM example, 168–170
busy waiting problem, 170–172
deadlocks, 167
dining philosophers problem, 175–178
monitors and semaphores and, 166–167
overview, 165–166
producer/consumer threads, 172–175
summary, 179
system threads vs. user thread, 166–167
three light switch problem, brainteasers, 265–266
three-valued logic, in SQL, 216–218
tie breakers, comparator function and, 157
token scanners, reversing order of words in string, 112–115
tracing garbage collectors, 297
train tunnel problem, graphical/spatial brainteasers, 276–277
training set, 245
traversal
of linked lists, 46
overview of, 81
preorder traversal, 84–87
tree rotation, balancing binary search trees, 92
trees
BFS (breadth-first search), 80–81
binary search trees, 78–80
converting binary tree to heap, 88–91
determining height of, 83–84
determining lowest common ancestor, 87–88
DFS (depth-first search), 81
heaps, 80
overview, 75–77
preorder traversal, 84–87
reorganizing an unbalanced binary search tree, 91–92
“Six Degrees of Kevin Bacon” game, 93–96
summary, 97
traversal, 81
tri-color marking, garbage collection and, 297
tuples (rows), relational database tables, 203

U
unary operators, 225
undirected graphs, 82
UNICODE
BMP (Basic Multilingual Plane) and, 108
characters and, 102
uniform distribution, 241, 246
unsupervised machine learning, 244
user interfaces, know your aptitude and likes, 1–2
user thread, vs. system threads, 166
UTF-8, 103, 121–123
UTF-16, 102
UX (user experience), know your aptitudes and likes, 1–2

V
validation set, 245
variables, sequence of. See arrays
Variance, 241
vertices, of graphs, 82
virtual methods, 186–188
visualization
as approach to solving graphical/spatial brainteasers, 276
cube counting problem, 278–282
cube counting problem, 278–282
cube counting problem, 278–282
cube counting problem, 278–282
cube counting problem, 278–282
Vocabulary, provided by design pattern, 194

W
weak references, garbage collection and, 296–297
Web application security, 298–300
websites, developing personal site, 7
WHERE clause, SQL statements, 207
word characters, distinguishing from nonword, 112
wrapper functions, use in recursion, 127
Wrapper pattern, 197

X
XOR operator, 225