



Table of Contents

Introduction	1
I The Interview	6
1 Getting Ready	7
2 Strategies For A Great Interview	12
3 Conducting An Interview	19
4 Problem Solving	23
II Problems	41
5 Primitive Types	42
5.1 Compute parity	42
5.2 Swap bits	44
5.3 Reverse bits	45
5.4 Find a closest integer with the same weight	46
5.5 Compute $x \times y$ without multiply or add	46
5.6 Compute x/y	47
5.7 Compute x^y	48
5.8 Convert base	49
5.9 Compute the spreadsheet column encoding	50
5.10 Reverse digits	50
5.11 Check if a decimal integer is a palindrome	51
5.12 Generate uniform random numbers	52
5.13 Check if rectangles intersect	53

6	Arrays	55
6.1	The Dutch national flag problem	55
6.2	Increment a BigInteger	57
6.3	Multiply two BigIntegers	58
6.4	Check if a board game is winnable	59
6.5	Delete a key from an array	60
6.6	Delete duplicates from a sorted array	61
6.7	Buy and sell a stock once	62
6.8	Buy and sell a stock twice	63
6.9	Enumerate all primes to n	63
6.10	Permute the elements of an array	65
6.11	Compute the next permutation	67
6.12	Sample offline data	69
6.13	Compute a random permutation	70
6.14	Compute a random subset of $\{0, 1, \dots, n - 1\}$	70
6.15	Sample online data	71
6.16	Generate nonuniform random numbers	72
6.17	The Sudoku checker problem	73
6.18	Compute the spiral ordering of a 2D array	75
6.19	Rotate a 2D array	77
6.20	Compute rows in Pascal's Triangle	78
7	Strings	80
7.1	Interconvert strings and integers	80
7.2	Replace and remove	81
7.3	Test palindromicity	83
7.4	Reverse all the words in a sentence	84
7.5	Compute all mnemonics for a phone number	85
7.6	The look-and-say problem	86
7.7	Convert from Roman to decimal	87
7.8	Compute all valid IP addresses	88
7.9	Write a string sinusoidally	90
7.10	Implement run-length encoding	90
7.11	Implement the UNIX <code>tail</code> command	91
7.12	Find the first occurrence of a substring	92
8	Linked Lists	95
8.1	Merge two sorted lists	96
8.2	Reverse a singly linked list	97
8.3	Reverse a single sublist	98
8.4	Test for cyclicity	99
8.5	Test for overlapping lists—lists are cycle-free	101
8.6	Test for overlapping lists—lists may have cycles	102
8.7	Delete a node from a singly linked list	104
8.8	Remove the k -th last element from a list	104

8.9	Remove duplicates from a sorted list	105
8.10	Implement cyclic right shift for singly linked lists	106
8.11	Implement even-odd merge	107
8.12	Test whether a singly linked list is palindromic	109
8.13	Implement list pivoting	110
8.14	Add list-based integers	111
9	Stacks and Queues	113
9.1	Implement a stack with max API	113
9.2	Evaluate RPN expressions	116
9.3	Test a string over “{ }, (), []” for well-formedness	117
9.4	Normalize pathnames	118
9.5	BST keys in sort order	120
9.6	Search a postings list	121
9.7	Compute buildings with a sunset view	122
9.8	Sort a stack	123
9.9	Compute binary tree nodes in order of increasing depth	125
9.10	Implement a circular queue	126
9.11	Implement a queue using stacks	127
9.12	Implement a queue with max API	128
10	Binary Trees	132
10.1	Test if a binary tree is balanced	134
10.2	Test if a binary tree is symmetric	135
10.3	Compute the lowest common ancestor in a binary tree	136
10.4	Compute the LCA when nodes have parent pointers	138
10.5	Sum the root-to-leaf paths in a binary tree	139
10.6	Find a root to leaf path with specified sum	140
10.7	Compute the k -th node in an inorder traversal	141
10.8	Implement an inorder traversal with $O(1)$ space	142
10.9	Compute the successor	143
10.10	Reconstruct a binary tree from traversal data	145
10.11	Reconstruct a binary tree from a preorder traversal with markers	147
10.12	Form a linked list from the leaves of a binary tree	148
10.13	Compute the exterior of a binary tree	148
10.14	Compute the right sibling tree	150
10.15	Implement locking in a binary tree	152
11	Heaps	154
11.1	Merge sorted files	155
11.2	Sort an increasing-decreasing array	156
11.3	Sort an almost-sorted array	157
11.4	Compute the k closest stars	158
11.5	Compute the median of online data	160
11.6	Compute the k largest elements in a max-heap	161

11.7	Implement a stack API using a heap	163
12	Searching	165
12.1	Search a sorted array for first occurrence of k	167
12.2	Search a sorted array for the first element greater than k	168
12.3	Search a sorted array for entry equal to its index	170
12.4	Search a cyclically sorted array	170
12.5	Compute the integer square root	172
12.6	Compute the real square root	173
12.7	Search in a 2D sorted array	175
12.8	Find the min and max simultaneously	176
12.9	Find the k -th largest element	178
12.10	Compute the optimum mailbox placement	179
12.11	Find the missing IP address	180
12.12	Find the duplicate and missing elements	182
13	Hash Tables	185
13.1	Partition into anagrams	186
13.2	Test for palindromic permutations	187
13.3	Is an anonymous letter constructible?	188
13.4	Implement an ISBN cache	189
13.5	Compute the LCA, optimizing for close ancestors	191
13.6	Compute the k most frequent queries	192
13.7	Find the nearest repeated entries in an array	193
13.8	Find the smallest subarray covering all values	194
13.9	Find smallest subarray that sequentially covering all values	197
13.10	Find the longest subarray with distinct entries	199
13.11	Find the length of a longest contained range	200
13.12	Compute the average of the top three scores	202
13.13	Compute all string decompositions	203
13.14	Find a highest affinity pair	205
13.15	Test the Collatz conjecture	206
13.16	Implement a hash function for chess	208
14	Sorting	210
14.1	Compute the intersection of two sorted arrays	211
14.2	Implement mergesort in-place	213
14.3	Count the frequencies of characters in a sentence	214
14.4	Find unique elements	215
14.5	Render a calendar	215
14.6	Sets of disjoint intervals	217
14.7	Compute the union of intervals	219
14.8	Partitioning and sorting an array with many repeated entries	221
14.9	Team photo day—1	223
14.10	Implement a fast sorting algorithm for lists	225

14.11	Compute a salary threshold	226
15	Binary Search Trees	228
15.1	Test if a binary tree satisfies the BST property	228
15.2	Find the first occurrence of a key in a BST	231
15.3	Find the first key larger than a given value in a BST	233
15.4	Find the k largest elements in a BST	234
15.5	Compute the LCA in a BST	235
15.6	Reconstruct a BST from traversal data	236
15.7	Find the closest entries in three sorted arrays	239
15.8	Enumerate numbers of the form $a + b\sqrt{2}$	240
15.9	The most visited pages problem	243
15.10	Build a BST from a sorted array	244
15.11	Insertion and deletion in a BST	245
15.12	Test if three BST nodes are totally ordered	248
15.13	The range lookup problem	250
15.14	Add credits	252
15.15	Count the number of entries in an interval	254
16	Recursion	256
16.1	The Tower of Hanoi problem	256
16.2	Generate all nonattacking placements of n -Queens	259
16.3	Generate permutations	261
16.4	Generate the power set	262
16.5	Generate all subsets of size k	264
16.6	Generate strings of matched parens	265
16.7	Generate palindromic decompositions	267
16.8	Generate binary trees	269
16.9	Implement a Sudoku solver	270
16.10	Compute a Gray code	272
16.11	Compute the diameter of a tree	274
17	Dynamic Programming	276
17.1	Count the number of score combinations	278
17.2	Compute the Levenshtein distance	279
17.3	Count the number of ways to traverse a 2D array	282
17.4	Plan a fishing trip	284
17.5	Search for a sequence in a 2D array	285
17.6	The knapsack problem	286
17.7	Divide the spoils fairly	287
17.8	The bedbathandbeyond.com problem	289
17.9	Find the maximum weight path in a triangle	291
17.10	Pick up coins for maximum gain	291
17.11	Count the number of steps in a board game	293
17.12	Compute the probability of a Republican majority	294

17.13	The pretty printing problem	296
17.14	Find the longest nondecreasing subsequence	297
18	Greedy Algorithms and Invariants	300
18.1	Implement Huffman coding	301
18.2	Compute an optimum assignment of tasks	304
18.3	Implement a schedule which minimizes waiting time	305
18.4	The interval covering problem	306
18.5	The 3-sum problem	309
18.6	Find the majority element	310
18.7	The gasup problem	311
18.8	Compute the maximum water trapped by a pair of vertical lines . .	312
18.9	Compute the largest rectangle under the skyline	314
19	Graphs	317
19.1	Identify the celebrity	319
19.2	Search a maze	320
19.3	Paint a Boolean matrix	322
19.4	Compute enclosed regions	324
19.5	Clone a graph	326
19.6	Making wired connections	327
19.7	Transform one string to another	329
19.8	The shortest straight-line program for x^n	330
19.9	Team photo day—2	332
19.10	Compute a shortest path with fewest edges	334
20	Parallel Computing	337
20.1	Implement caching for a multithreaded dictionary	338
20.2	Analyze two unsynchronized interleaved threads	340
20.3	Implement synchronization for two interleaving threads	341
20.4	Implement a thread pool	343
20.5	Implement asynchronous callbacks	344
20.6	Implement a Timer class	345
20.7	The readers-writers problem	346
20.8	The readers-writers problem with write preference	347
20.9	Test the Collatz conjecture in parallel	348
20.10	Design TeraSort and PetaSort	350
20.11	Implement distributed throttling	351
21	Design Problems	352
21.1	Design a spell checker	354
21.2	Design a solution to the stemming problem	354
21.3	Plagiarism detector	355
21.4	Pair users by attributes	356
21.5	Design a system for detecting copyright infringement	357

21.6	Design \TeX	358
21.7	Design a search engine	359
21.8	Implement PageRank	360
21.9	Design a scalable priority system	361
21.10	Create photomosaics	362
21.11	Implement Mileage Run	362
21.12	Implement Connexus	364
21.13	Design an online advertising system	364
21.14	Design a recommendation system	365
21.15	Design an optimized way of distributing large files	366
21.16	Design the World Wide Web	367
21.17	Estimate the hardware cost of a photo sharing app	368

22 Honors Class 369

22.1	Compute the greatest common divisor 🧐	370
22.2	Find the first missing positive entry 🧐	371
22.3	Buy and sell a stock k times 🧐	372
22.4	Compute the maximum product of all but one entries 🧐	373
22.5	Compute the longest contiguous increasing subarray 🧐	376
22.6	Rotate an array 🧐	377
22.7	Identify positions attacked by rooks 🧐	379
22.8	Justify text 🧐	381
22.9	Reverse sublists k at a time 🧐	382
22.10	Implement list zipping 🧐	383
22.11	Copy a postings list 🧐	384
22.12	Compute the median of a sorted circular linked list 🧐	386
22.13	Compute the longest substring with matching parens 🧐	387
22.14	Compute the maximum of a sliding window 🧐	388
22.15	Implement preorder and postorder traversals without recursion 🧐	390
22.16	Compute fair bonuses 🧐	393
22.17	Find k elements closest to the median 🧐	395
22.18	Search a sorted array of unknown length 🧐	397
22.19	Search in two sorted arrays 🧐	398
22.20	Find the k -th largest element—large n , small k 🧐	400
22.21	Find an element that appears only once 🧐	401
22.22	Find the line through the most points 🧐	402
22.23	Find the shortest unique prefix 🧐	405
22.24	Compute the smallest nonconstructible change 🧐	407
22.25	Find the most visited pages in a window 🧐	408
22.26	Convert a sorted doubly linked list into a BST 🧐	409
22.27	Convert a BST to a sorted doubly linked list 🧐	411
22.28	Merge two BSTs 🧐	412
22.29	Test if a binary tree is an almost BST 🧐	414
22.30	The view from above 🧐	416

22.31	Searching a min-first BST 🧐	418
22.32	Implement regular expression matching 🧐	420
22.33	Synthesize an expression 🧐	423
22.34	Count inversions 🧐	425
22.35	Draw the skyline 🧐	427
22.36	Find the two closest points 🧐	430
22.37	Measure with defective jugs 🧐	433
22.38	Compute the maximum subarray sum in a circular array 🧐	434
22.39	Determine the critical height 🧐	436
22.40	Voltage selection in a logic circuit 🧐	438
22.41	Find the maximum 2D subarray 🧐	439
22.42	Trapping water 🧐	442
22.43	Load balancing 🧐	443
22.44	Search for a pair-sum in an abs-sorted array 🧐	445
22.45	The heavy hitter problem 🧐	448
22.46	Find the longest subarray whose sum $\leq k$ 🧐	449
22.47	Test degrees of connectedness 🧐	451
22.48	Compute a minimum delay schedule, unlimited resources 🧐	454
22.49	Road network 🧐	454
22.50	Test if arbitrage is possible 🧐	456
22.51	The readers-writers problem with fairness 🧐	458
22.52	Implement a producer-consumer queue 🧐	459

III Notation, Language, and Index 460

Notation 461

C++11, and C++ for Java developers 463

Index of Terms 465