

Index

- 2-3-trees, 98
- access rights, 115
- Ackermann's function, 126
- Adelson-Velsky, G., 65
- affix code, 202
- algorithm
 - Boyer–Moore, 1–12, 116
 - Dijkstra, 47
 - Heapsort, 110
 - Karp–Rabin, 148
 - Knuth–Morris–Pratt, 2
 - Kruskal, 126
 - Mergesort, 26
 - probabilistic, 148
- allocation
 - linked, 16
 - sequential, 15
- alphabet, 4
- amortized analysis, 89
- ASCII, 129, 132, 179
- asymptotic notation, 20
- AVL tree, 65–82, 86, 159
 - balance factor, 71
 - deletion, 77
 - depth, 66
 - insertion, 71
 - rotation, 75
- B*-trees, 96–98
 - deletion, 97
 - insertion, 96
- B⁺-trees, 93–95, 156
 - deletion, 95
 - insertion, 94
 - search, 95
- B-tree, 83–100, 156
 - deletion, 90
 - depth, 86
 - insertion, 86
- background concept
 - approximation of a sum, 144
 - asymptotic notations, 20
 - average complexity, 161
 - binary search, 15
 - birthday paradox, 135
 - depth of a tree, 65
 - Fibonacci sequence, 70
 - mergesort, 26
 - modular arithmetic, 133
 - number of binary trees, 53
 - prime numbers, 131
 - proving properties of trees, 67
 - recurrence relations, 175
 - summation, 19, 142
 - swapping two elements, 108
- BFS, *see* search
- big-O, 20
- binary representation, 133
- binary search, 15
- binary search tree, 58
- binary tree, 52, 182
 - complete, 56
 - full, 56, 103
 - number, 53
- birthday, 135
- bitmap, 114
 - compression, 41

- Bloom filters, 149
- Boyer, J S., 1
- bubblesort, 153, 157
- C programming language, 132
- Caesar's code, 201
- Catalan numbers, 54
- chaining, 137
- character frequency, 179
- chmod command, 115
- ciphertext, 200
- circular list, 29
- cleartext, 200
- cluster, 138
- code
 - affix, 202
 - breaking, 201
 - complete, 183, 202
 - compression, 179
 - cryptographic, 179
 - error-correcting, 179
 - fixed length, 178
 - Hamming, 197
 - robustness, 194
 - unary, 190
 - universal, 179
 - variable length, 45, 180
- codes, 178–203
- complete code, 183
- complexity, 5, 161
- compression
 - bitmaps, 41
 - data, 129, 148
- concordance, 116
- counting sort, 157
- cryptography, 199
- De Bruijn sequence, 48
- decision tree, 158
- deduplication, 148
- δ -code, 190
- Δ_1 , 5
- Δ_2 , 7
- depth
 - node, 57
- deque, 30
- derivative, 165
- DFS, *see* search
- dictionary, 44, 116, 189
- Dijkstra, E. W., 47
- divide and conquer, 15, 26, 166
- doubly linked list, 29
- edge, 33
- Elias code, 190
- Elias, P., 190
- entropy, 185, 202
- error-correcting code, 199
- error correction, 194
- error detection, 195
- expectation, 141, 161
- expression
 - arithmetic, 22, 56
- extraction
 - queue, 17
- Fibonacci
 - code, 193
 - numbers, 81
 - sequence, 70
- fields, 14
- FIFO, 17
- game of Hex, 125
- γ -code, 190
- geometric progression,
 - 142
- geometric series, 110
- golden ratio, 70
- graph, 33–49
 - exploration, 40
 - planar, 208
 - representation, 38
 - tree, 35
- greedy, 44
- Hamming code, 197
- Hamming distance, 43
- Hamming, R., 197
- harmonic series, 144, 170
- hashing, 127–151
 - deletion, 147
 - double, 139
 - insertion, 141
 - search, 143
 - uniform, 140, 151, 170
 - worst case, 150
- heaps, 101–113, 154
 - array implementation, 105
 - construction, 106
 - max-heap, 112
 - min-heap, 112
 - ternary, 113
- heapsort, 110, 153, 154, 166, 196

- heuristic
 - bad-character, 4
 - good-suffix, 7
- Horner's rule, 134
- Huffman
 - higher-order, 203
- Huffman code
 - decompression, 186
 - optimality, 187
- Huffman coding, 183–189
- Huffman, D., 185
- in-order, 55
- induction, 57
- infix, 23
- information content, 184
- information retrieval, 41, 115, 149, 192
- insertion
 - queue, 17
- insertion sort, 153
- instantaneous code, 181
- inverted files, 116
- k*-largest element, 170
- k*-layer sort, 154
- Karp, R. M., 148
- Kruskal, J. B., 126
- Landis, E., 65
- Lempel, A., 190
- LIFO, 21
- linear lists, 14–32, 33
 - cycle, 206
- linked list, 16
- list
 - neighbors, 39
- load factor, 141
- \log^* , 124, 126
- lower bound, 156
- matrix, 30
 - adjacency, 38, 115
 - sparse, 31
- maximum flow, 37, 47
- median, 170
- mergesort, 26, 153, 157, 166
- merging
 - sequences, 64
 - trees, 64
- middle square method, 129
- minimum spanning tree, 36, 43, 126
- modulus, 133
- Moore, R. S., 1
- natural logarithm, 146
- Nobel laureates, 136, 185
- node
 - balancing, 90
 - merging, 91
 - splitting, 87
- notation
 - asymptotic, 20
 - Polish, 22
 - reversed Polish, 23
- Ω , 21
- open addressing, 137
- operand, 56
- operator, 56
- overflow, 17
- parity bit, 195
- parsing, 44
- password, 201
- path compression, 123
- π , 134, 151, 153
- pop, 22
- post-order, 55
- postfix, 23
- pre-order, 55
- prefix, 23
- prefix code, 18, 181
- prefix property, 181
- prime
 - twin, 140
- prime number, 131, 199
- primitive function, 146
- priority, 24
- priority queue, 101
- probe sequence, 138
- problem
 - four color, 48
- program stack, 25
- pseudo random number, 129
- push, 22
- query, 115
 - range, 150
- queue, 16, 41
 - extraction, 17
 - insertion, 17
- quicksort, 157, 166–170
 - worst case, 168, 169

- règle par neuf*, 151
- Rabin, M. O., 148
- radix sort, 157
- records, 14
- recursion, 25
- reduction, 36
- remainder, 130
- search, 126
 - approximate, 150
 - binary, 15
 - breadth first (BFS), 40
 - depth first (DFS), 40
- search tree
 - binary, 58, 123
 - deletion, 61
 - higher-order, 83
 - insertion, 60
 - search, 59
- selection sort, 153, 196
- sentinel, 24, 29, 63, 167
- sequence
 - De Bruijn, 48
 - Fibonacci, 70
- set, 114–126
 - merging, 118
- Shannon, C. E., 185
- shortest path, 36, 45
- signature files, 149
- singleton, 115, 158
- sorting, 152–177
 - average, 160
 - worst case, 156
- stack, 21, 41
- string, 1
- string matching, 1, 116, 148
- summation, 19
- swap, 108, 167
- text, 1
- θ , 21
- tree
 - AVL, *see* AVL trees
 - data structure, 50–64
 - decision, 158
 - depth, 65
 - graph, 35
 - height-balanced, 66
 - red-black, 80
 - rooted, 118
 - ternary, 64, 113
- twin prime, 131
- UD, *see* uniquely decipherable
- Union-Find, 117–124
- uniquely decipherable, 180
- universal codes, 189–194
- Unix, 115
- vertex, 33
- Vigenère, B., 201
- von Neumann, J., 129
- xor, 42, 108, 130, 195
- Ziv, J., 190