1. [Heap Sort](http://geeksquiz.com/heap-sort/), [QuickSort](http://geeksquiz.com/quick-sort/" \t "_blank), , [Pigeonhole Sort](https://www.geeksforgeeks.org/pigeonhole-sort/), [Merge Sort for Linked Lists](https://www.geeksforgeeks.org/merge-sort-for-linked-list/), [Iterative Quick Sort](https://www.geeksforgeeks.org/iterative-quick-sort/), [QuickSort on Singly Linked List](https://www.geeksforgeeks.org/quicksort-on-singly-linked-list/), [QuickSort on Doubly Linked List](https://www.geeksforgeeks.org/quicksort-for-linked-list/), [Binary Insertion Sort](http://geeksquiz.com/binary-insertion-sort/), [Insertion Sort for Singly Linked List](http://geeksquiz.com/insertion-sort-for-singly-linked-list/), [Merge Sort for Doubly Linked List](https://www.geeksforgeeks.org/merge-sort-for-doubly-linked-list/)
2. [Find k closest elements to a given value](https://www.geeksforgeeks.org/find-k-closest-elements-given-value/)
3. [Sort n numbers in range from 0 to n^2 – 1 in linear time](https://www.geeksforgeeks.org/sort-n-numbers-range-0-n2-1-linear-time/)
4. [K’th Smallest/Largest Element in Unsorted Array](https://www.geeksforgeeks.org/kth-smallestlargest-element-unsorted-array-set-2-expected-linear-time/)
5. [K’th Smallest/Largest Element in Unsorted Array in Expected Linear Time](https://www.geeksforgeeks.org/kth-smallestlargest-element-unsorted-array-set-2-expected-linear-time/)
6. [K’th Smallest/Largest Element in Unsorted Array in Worst Case Linear Time](https://www.geeksforgeeks.org/kth-smallestlargest-element-unsorted-array-set-3-worst-case-linear-time/)
7. [Find the closest pair from two sorted arrays](https://www.geeksforgeeks.org/given-two-sorted-arrays-number-x-find-pair-whose-sum-closest-x/)
8. [Find common elements in three sorted arrays](https://www.geeksforgeeks.org/find-common-elements-three-sorted-arrays/)
9. [Given a sorted array and a number x, find the pair in array whose sum is closest to x](http://geeksquiz.com/given-sorted-array-number-x-find-pair-array-whose-sum-closest-x/)
10. [Count 1’s in a sorted binary array](http://geeksquiz.com/count-1s-sorted-binary-array/" \o "Permanent link to Count 1’s in a sorted binary array)
11. [Minimum adjacent swaps to move maximum and minimum to corners](https://www.geeksforgeeks.org/minimum-adjacent-swaps-to-move-maximum-and-minimum-to-corners/" \t "_blank)

**Greedy Algorithms**:

1. [Activity Selection Problem](https://www.geeksforgeeks.org/greedy-algorithms-set-1-activity-selection-problem/)
2. [Kruskal’s Minimum Spanning Tree Algorithm](https://www.geeksforgeeks.org/greedy-algorithms-set-2-kruskals-minimum-spanning-tree-mst/)
3. [Huffman Coding](https://www.geeksforgeeks.org/greedy-algorithms-set-3-huffman-coding/)
4. [Efficient Huffman Coding for Sorted Input](https://www.geeksforgeeks.org/greedy-algorithms-set-3-huffman-coding-set-2/)
5. [Prim’s Minimum Spanning Tree Algorithm](https://www.geeksforgeeks.org/greedy-algorithms-set-5-prims-minimum-spanning-tree-mst-2/)
6. [Prim’s MST for Adjacency List Representation](https://www.geeksforgeeks.org/greedy-algorithms-set-5-prims-mst-for-adjacency-list-representation/)
7. [Dijkstra’s Shortest Path Algorithm](https://www.geeksforgeeks.org/greedy-algorithms-set-6-dijkstras-shortest-path-algorithm/)
8. [Dijkstra’s Algorithm for Adjacency List Representation](https://www.geeksforgeeks.org/greedy-algorithms-set-7-dijkstras-algorithm-for-adjacency-list-representation/)
9. [Job Sequencing Problem](https://www.geeksforgeeks.org/job-sequencing-problem-set-1-greedy-algorithm/)
10. [Greedy Algorithm to find Minimum number of Coins](http://geeksquiz.com/greedy-algorithm-to-find-minimum-number-of-coins/)
11. [K Centers Problem](https://www.geeksforgeeks.org/k-centers-problem-set-1-greedy-approximate-algorithm/)
12. [Minimum Number of Platforms Required for a Railway/Bus Station](https://www.geeksforgeeks.org/minimum-number-platforms-required-railwaybus-station/)

**Dynamic Programming**:

1. [Overlapping Subproblems Property](https://www.geeksforgeeks.org/dynamic-programming-set-1/)
2. [Optimal Substructure Property](https://www.geeksforgeeks.org/dynamic-programming-set-2-optimal-substructure-property/)
3. [Longest Increasing Subsequence](https://www.geeksforgeeks.org/dynamic-programming-set-3-longest-increasing-subsequence/)
4. [Longest Common Subsequence](https://www.geeksforgeeks.org/dynamic-programming-set-4-longest-common-subsequence/)
5. [Edit Distance](https://www.geeksforgeeks.org/dynamic-programming-set-5-edit-distance/)
6. [Min Cost Path](https://www.geeksforgeeks.org/dynamic-programming-set-6-min-cost-path/)
7. [Coin Change](https://www.geeksforgeeks.org/dynamic-programming-set-7-coin-change/)
8. [Matrix Chain Multiplication](https://www.geeksforgeeks.org/dynamic-programming-set-8-matrix-chain-multiplication/)
9. [Binomial Coefficient](https://www.geeksforgeeks.org/dynamic-programming-set-9-binomial-coefficient/)
10. [0-1 Knapsack Problem](https://www.geeksforgeeks.org/dynamic-programming-set-10-0-1-knapsack-problem/)
11. [Egg Dropping Puzzle](https://www.geeksforgeeks.org/dynamic-programming-set-11-egg-dropping-puzzle/)
12. [Longest Palindromic Subsequence](https://www.geeksforgeeks.org/dynamic-programming-set-12-longest-palindromic-subsequence/)
13. [Cutting a Rod](https://www.geeksforgeeks.org/dynamic-programming-set-13-cutting-a-rod/)
14. [Maximum Sum Increasing Subsequence](https://www.geeksforgeeks.org/dynamic-programming-set-14-maximum-sum-increasing-subsequence/)
15. [Longest Bitonic Subsequence](https://www.geeksforgeeks.org/dynamic-programming-set-15-longest-bitonic-subsequence/)
16. [Floyd Warshall Algorithm](https://www.geeksforgeeks.org/dynamic-programming-set-16-floyd-warshall-algorithm/)
17. [Palindrome Partitioning](https://www.geeksforgeeks.org/dynamic-programming-set-17-palindrome-partitioning/)
18. [Partition problem](https://www.geeksforgeeks.org/dynamic-programming-set-18-partition-problem/)
19. [Word Wrap Problem](https://www.geeksforgeeks.org/dynamic-programming-set-18-word-wrap/)
20. [Maximum Length Chain of Pairs](https://www.geeksforgeeks.org/dynamic-programming-set-20-maximum-length-chain-of-pairs/)
21. [Variations of LIS](https://www.geeksforgeeks.org/dynamic-programming-set-14-variations-of-lis/)
22. [Box Stacking Problem](https://www.geeksforgeeks.org/dynamic-programming-set-21-box-stacking-problem/)
23. [Program for Fibonacci numbers](https://www.geeksforgeeks.org/program-for-nth-fibonacci-number/)
24. [Minimum number of jumps to reach end](https://www.geeksforgeeks.org/minimum-number-of-jumps-to-reach-end-of-a-given-array/)
25. [Maximum size square sub-matrix with all 1s](https://www.geeksforgeeks.org/maximum-size-sub-matrix-with-all-1s-in-a-binary-matrix/)
26. [Ugly Numbers](https://www.geeksforgeeks.org/ugly-numbers/)
27. [Largest Sum Contiguous Subarray](https://www.geeksforgeeks.org/largest-sum-contiguous-subarray/)
28. [Longest Palindromic Substring](https://www.geeksforgeeks.org/longest-palindrome-substring-set-1/)
29. [Bellman–Ford Algorithm for Shortest Paths](https://www.geeksforgeeks.org/dynamic-programming-set-23-bellman-ford-algorithm/)

**Pattern Searching:**

1. [Naive Pattern Searching](https://www.geeksforgeeks.org/searching-for-patterns-set-1-naive-pattern-searching/)
2. [KMP Algorithm](https://www.geeksforgeeks.org/searching-for-patterns-set-2-kmp-algorithm/)
3. [Rabin-Karp Algorithm](https://www.geeksforgeeks.org/searching-for-patterns-set-3-rabin-karp-algorithm/)
4. [A Naive Pattern Searching Question](https://www.geeksforgeeks.org/pattern-searching-set-4-a-naive-string-matching-algo-question/)
5. [Finite Automata](https://www.geeksforgeeks.org/searching-for-patterns-set-5-finite-automata/)
6. [Efficient Construction of Finite Automata](https://www.geeksforgeeks.org/pattern-searching-set-5-efficient-constructtion-of-finite-automata/)
7. [Boyer Moore Algorithm – Bad Character Heuristic](https://www.geeksforgeeks.org/pattern-searching-set-7-boyer-moore-algorithm-bad-character-heuristic/)
8. [Suffix Array](https://www.geeksforgeeks.org/suffix-array-set-1-introduction/)
9. [Anagram Substring Search (Or Search for all permutations)](https://www.geeksforgeeks.org/anagram-substring-search-search-permutations/)
10. [Pattern Searching using a Trie of all Suffixes](https://www.geeksforgeeks.org/pattern-searching-using-trie-suffixes/)
11. [Aho-Corasick Algorithm for Pattern Searching](https://www.geeksforgeeks.org/aho-corasick-algorithm-pattern-searching/)
12. [kasai’s Algorithm for Construction of LCP array from Suffix Array](https://www.geeksforgeeks.org/%c2%ad%c2%adkasais-algorithm-for-construction-of-lcp-array-from-suffix-array/)
13. [Z algorithm (Linear time pattern searching Algorithm)](https://www.geeksforgeeks.org/z-algorithm-linear-time-pattern-searching-algorithm/)

**Other String Algorithms:**

1. [Manacher’s Algorithm – Linear Time Longest Palindromic Substring – Part 1](https://www.geeksforgeeks.org/manachers-algorithm-linear-time-longest-palindromic-substring-part-1/), [Part 2](https://www.geeksforgeeks.org/manachers-algorithm-linear-time-longest-palindromic-substring-part-2/), [Part 3](https://www.geeksforgeeks.org/manachers-algorithm-linear-time-longest-palindromic-substring-part-3-2/), [Part 4](https://www.geeksforgeeks.org/manachers-algorithm-linear-time-longest-palindromic-substring-part-4/)
2. [Longest Even Length Substring such that Sum of First and Second Half is same](https://www.geeksforgeeks.org/longest-even-length-substring-sum-first-second-half/)
3. [Print all possible strings that can be made by placing spaces](https://www.geeksforgeeks.org/print-possible-strings-can-made-placing-spaces/)

**Backtracking**:

1. [Print all permutations of a given string](https://www.geeksforgeeks.org/write-a-c-program-to-print-all-permutations-of-a-given-string/)
2. [The Knight’s tour problem](https://www.geeksforgeeks.org/backtracking-set-1-the-knights-tour-problem/)
3. [Rat in a Maze](https://www.geeksforgeeks.org/backttracking-set-2-rat-in-a-maze/)
4. [N Queen Problem](https://www.geeksforgeeks.org/backtracking-set-3-n-queen-problem/)
5. [Subset Sum](https://www.geeksforgeeks.org/backttracking-set-4-subset-sum/)
6. [m Coloring Problem](https://www.geeksforgeeks.org/backttracking-set-5-m-coloring-problem/)
7. [Hamiltonian Cycle](https://www.geeksforgeeks.org/backtracking-set-7-hamiltonian-cycle/)
8. [Sudoku](https://www.geeksforgeeks.org/backtracking-set-7-suduku/)
9. [Tug of War](https://www.geeksforgeeks.org/tug-of-war/)
10. [Solving Cryptarithmetic Puzzles](https://www.geeksforgeeks.org/backtracking-set-8-solving-cryptarithmetic-puzzles/)

**Divide and Conquer**:

1. [Write your own pow(x, n) to calculate x\*n](https://www.geeksforgeeks.org/write-a-c-program-to-calculate-powxn/)
2. [Median of two sorted arrays](https://www.geeksforgeeks.org/median-of-two-sorted-arrays/)
3. [Count Inversions](https://www.geeksforgeeks.org/counting-inversions/)
4. [Closest Pair of Points](https://www.geeksforgeeks.org/closest-pair-of-points/)
5. [Strassen’s Matrix Multiplication](https://www.geeksforgeeks.org/strassens-matrix-multiplication/)

**Geometric Algorithms:**

1. [Closest Pair of Points | O(nlogn) Implementation](https://www.geeksforgeeks.org/closest-pair-of-points-onlogn-implementation/)
2. [How to check if two given line segments intersect?](https://www.geeksforgeeks.org/check-if-two-given-line-segments-intersect/)
3. [How to check if a given point lies inside or outside a polygon?](https://www.geeksforgeeks.org/how-to-check-if-a-given-point-lies-inside-a-polygon/)
4. [Convex Hull | Set 1 (Jarvis’s Algorithm or Wrapping)](https://www.geeksforgeeks.org/convex-hull-set-1-jarviss-algorithm-or-wrapping/)
5. [Convex Hull | Set 2 (Graham Scan)](https://www.geeksforgeeks.org/convex-hull-set-2-graham-scan/)
6. [Given n line segments, find if any two segments intersect](https://www.geeksforgeeks.org/given-a-set-of-line-segments-find-if-any-two-segments-intersect/)
7. [Check whether a given point lies inside a triangle or not](https://www.geeksforgeeks.org/check-whether-a-given-point-lies-inside-a-triangle-or-not/)
8. [How to check if given four points form a square](http://geeksquiz.com/check-given-four-points-form-square/)

**Mathematical Algorithms:**

1. [Write an Efficient Method to Check if a Number is Multiple of 3](https://www.geeksforgeeks.org/write-an-efficient-method-to-check-if-a-number-is-multiple-of-3/)
2. [Efficient way to multiply with 7](https://www.geeksforgeeks.org/efficient-way-to-multiply-with-7/)
3. [Write a C program to print all permutations of a given string](https://www.geeksforgeeks.org/write-a-c-program-to-print-all-permutations-of-a-given-string/)
4. [Lucky Numbers](https://www.geeksforgeeks.org/lucky-numbers/)
5. [Write a program to add two numbers in base 14](https://www.geeksforgeeks.org/write-a-program-to-add-two-numbers-in-base-14/)
6. [Babylonian method for square root](https://www.geeksforgeeks.org/square-root-of-a-perfect-square/)
7. [Write you own Power without using multiplication(\*) and division(/) operators](https://www.geeksforgeeks.org/write-you-own-power-without-using-multiplication-and-division/)
8. [Program for Fibonacci numbers](https://www.geeksforgeeks.org/program-for-nth-fibonacci-number/)
9. [Average of a stream of numbers](https://www.geeksforgeeks.org/average-of-a-stream-of-numbers/)
10. [Count numbers that don’t contain 3](https://www.geeksforgeeks.org/count-numbers-that-dont-contain-3/)
11. [MagicSquare](https://www.geeksforgeeks.org/magic-square/)
12. [Sieve of Eratosthenes](https://www.geeksforgeeks.org/sieve-of-eratosthenes/)
13. [Number which has the maximum number of distinct prime factors in the range M to N](https://www.geeksforgeeks.org/number-which-has-the-maximum-number-of-distinct-prime-factors-in-range-m-to-n/)
14. [Find day of the week for a given date](https://www.geeksforgeeks.org/find-day-of-the-week-for-a-given-date/)
15. [DFA based division](https://www.geeksforgeeks.org/dfa-based-division/)
16. [Generate integer from 1 to 7 with equal probability](https://www.geeksforgeeks.org/generate-integer-from-1-to-7-with-equal-probability/)
17. [Given a number, find the next smallest palindrome](https://www.geeksforgeeks.org/given-a-number-find-next-smallest-palindrome-larger-than-this-number/)
18. [Make a fair coin from a biased coin](https://www.geeksforgeeks.org/print-0-and-1-with-50-probability/)
19. [Check divisibility by 7](https://www.geeksforgeeks.org/divisibility-by-7/)
20. [Find the largest multiple of 3](https://www.geeksforgeeks.org/find-the-largest-number-multiple-of-3/)
21. [Lexicographic rank of a string](https://www.geeksforgeeks.org/lexicographic-rank-of-a-string/)
22. [Print all permutations in sorted (lexicographic) order](https://www.geeksforgeeks.org/lexicographic-permutations-of-string/)
23. [Shuffle a given array](https://www.geeksforgeeks.org/shuffle-a-given-array/)
24. [Space and time efficient Binomial Coefficient](https://www.geeksforgeeks.org/space-and-time-efficient-binomial-coefficient/)
25. [Reservoir Sampling](https://www.geeksforgeeks.org/reservoir-sampling/)
26. [Select a random number from stream, with O(1) space](https://www.geeksforgeeks.org/select-a-random-number-from-stream-with-o1-space/)
27. [Find the largest multiple of 2, 3 and 5](https://www.geeksforgeeks.org/find-the-largest-multiple-of-2-3-and-5/)
28. [Efficient program to calculate e^x](https://www.geeksforgeeks.org/program-to-efficiently-calculate-ex/)
29. [Measure one litre using two vessels and infinite water supply](https://www.geeksforgeeks.org/measure-1-litre-from-two-vessels-infinite-water-supply/)
30. [Tower of Hanoi](http://geeksquiz.com/c-program-for-tower-of-hanoi/)

**Bit Algorithms:**

1. [Find the element that appears once](https://www.geeksforgeeks.org/find-the-element-that-appears-once/)
2. [Detect opposite signs](https://www.geeksforgeeks.org/detect-if-two-integers-have-opposite-signs/)
3. [Set bits in all numbers from 1 to n](https://www.geeksforgeeks.org/count-total-set-bits-in-all-numbers-from-1-to-n/)
4. [Swap bits](https://www.geeksforgeeks.org/swap-bits-in-a-given-number/)
5. [Add two numbers](https://www.geeksforgeeks.org/add-two-numbers-without-using-arithmetic-operators/)
6. [Smallest of three](https://www.geeksforgeeks.org/smallest-of-three-integers-without-comparison-operators/)
7. [A Boolean Array Puzzle](https://www.geeksforgeeks.org/a-boolean-array-puzzle/)
8. [Set bits in an (big) array](https://www.geeksforgeeks.org/program-to-count-number-of-set-bits-in-an-big-array/)
9. [Next higher number with same number of set bits](https://www.geeksforgeeks.org/next-higher-number-with-same-number-of-set-bits/)
10. [Optimization Technique (Modulus)](https://www.geeksforgeeks.org/optimization-techniques-set-1-modulus/)
11. [Multiply with 3.5](https://www.geeksforgeeks.org/multiply-an-integer-with-3-5/)
12. [Turn off the rightmost set bit](https://www.geeksforgeeks.org/turn-off-the-rightmost-set-bit/)
13. [Check for Power of 4](https://www.geeksforgeeks.org/find-whether-a-given-number-is-a-power-of-4-or-not/)
14. [Absolute value (abs) without branching](https://www.geeksforgeeks.org/compute-the-integer-absolute-value-abs-without-branching/)
15. [Modulus division by a power-of-2-number](https://www.geeksforgeeks.org/compute-modulus-division-by-a-power-of-2-number/)
16. [Minimum or Maximum of two integers](https://www.geeksforgeeks.org/compute-the-minimum-or-maximum-max-of-two-integers-without-branching/)
17. [Rotate bits](https://www.geeksforgeeks.org/rotate-bits-of-an-integer/)
18. [Find the two non-repeating elements in an array](https://www.geeksforgeeks.org/find-two-non-repeating-elements-in-an-array-of-repeating-elements/)
19. [Number Occurring Odd Number of Times](https://www.geeksforgeeks.org/find-the-number-occurring-odd-number-of-times/)
20. [Karatsuba algorithm for fast multiplication](https://www.geeksforgeeks.org/divide-and-conquer-set-2-karatsuba-algorithm-for-fast-multiplication/)

**Graph Algorithms:**

***Introduction, DFS and BFS:***

1. [Breadth First Traversal for a Graph](https://www.geeksforgeeks.org/breadth-first-traversal-for-a-graph/" \o "Permanent link to Breadth First Traversal for a Graph)
2. [Depth First Traversal for a Graph](https://www.geeksforgeeks.org/depth-first-traversal-for-a-graph/)
3. [Applications of Depth First Search](https://www.geeksforgeeks.org/applications-of-depth-first-search/)
4. [Detect Cycle in a Directed Graph](https://www.geeksforgeeks.org/detect-cycle-in-a-graph/)
5. [Detect Cycle in an Undirected Graph](https://www.geeksforgeeks.org/union-find/)
6. [Longest Path in a Directed Acyclic Graph](https://www.geeksforgeeks.org/find-longest-path-directed-acyclic-graph/)
7. [Topological Sorting](https://www.geeksforgeeks.org/topological-sorting/)
8. [Check whether a given graph is Bipartite or not](https://www.geeksforgeeks.org/bipartite-graph/)
9. [Snake and Ladder Problem](https://www.geeksforgeeks.org/snake-ladder-problem-2/)
10. [Biconnected Components](https://www.geeksforgeeks.org/biconnected-components/)
11. [Check if a given graph is tree or not](http://geeksquiz.com/check-given-graph-tree/)

***Minimum Spanning Tree:***

1. [Prim’s Minimum Spanning Tree (MST))](https://www.geeksforgeeks.org/greedy-algorithms-set-5-prims-minimum-spanning-tree-mst-2/)
2. [Applications of Minimum Spanning Tree Problem](https://www.geeksforgeeks.org/applications-of-minimum-spanning-tree/)
3. [Prim’s MST for Adjacency List Representation](https://www.geeksforgeeks.org/greedy-algorithms-set-5-prims-mst-for-adjacency-list-representation/)
4. [Kruskal’s Minimum Spanning Tree Algorithm](https://www.geeksforgeeks.org/greedy-algorithms-set-2-kruskals-minimum-spanning-tree-mst/)
5. [Boruvka’s algorithm for Minimum Spanning Tree](https://www.geeksforgeeks.org/greedy-algorithms-set-9-boruvkas-algorithm/)

***Shortest Paths:***

1. [Dijkstra’s shortest path algorithm](https://www.geeksforgeeks.org/greedy-algorithms-set-6-dijkstras-shortest-path-algorithm/)
2. [Dijkstra’s Algorithm for Adjacency List Representation](https://www.geeksforgeeks.org/greedy-algorithms-set-7-dijkstras-algorithm-for-adjacency-list-representation/)
3. [Bellman–Ford Algorithm](https://www.geeksforgeeks.org/dynamic-programming-set-23-bellman-ford-algorithm/)
4. [Floyd Warshall Algorithm](https://www.geeksforgeeks.org/dynamic-programming-set-16-floyd-warshall-algorithm/)
5. [Johnson’s algorithm for All-pairs shortest paths](https://www.geeksforgeeks.org/johnsons-algorithm/)
6. [Shortest Path in Directed Acyclic Graph](https://www.geeksforgeeks.org/shortest-path-for-directed-acyclic-graphs/)
7. [Some interesting shortest path questions](https://www.geeksforgeeks.org/interesting-shortest-path-questions-set-1/)
8. [Shortest path with exactly k edges in a directed and weighted graph](https://www.geeksforgeeks.org/shortest-path-exactly-k-edges-directed-weighted-graph/)

***Connectivity:***

1. [Find if there is a path between two vertices in a directed graph](https://www.geeksforgeeks.org/find-if-there-is-a-path-between-two-vertices-in-a-given-graph/)
2. [Connectivity in a directed graph](https://www.geeksforgeeks.org/connectivity-in-a-directed-graph/)
3. [Articulation Points (or Cut Vertices) in a Graph](https://www.geeksforgeeks.org/articulation-points-or-cut-vertices-in-a-graph/)
4. [Biconnected graph](https://www.geeksforgeeks.org/biconnectivity-in-a-graph/)
5. [Bridges in a graph](https://www.geeksforgeeks.org/bridge-in-a-graph/)
6. [Eulerian path and circuit](https://www.geeksforgeeks.org/eulerian-path-and-circuit/)
7. [Fleury’s Algorithm for printing Eulerian Path or Circuit](https://www.geeksforgeeks.org/fleurys-algorithm-for-printing-eulerian-path/)
8. [Strongly Connected Components](https://www.geeksforgeeks.org/strongly-connected-components/)
9. [Transitive closure of a graph](https://www.geeksforgeeks.org/transitive-closure-of-a-graph/)
10. [Find the number of islands](https://www.geeksforgeeks.org/find-number-of-islands/)
11. [Count all possible walks from a source to a destination with exactly k edges](https://www.geeksforgeeks.org/count-possible-paths-source-destination-exactly-k-edges/)
12. [Euler Circuit in a Directed Graph](https://www.geeksforgeeks.org/euler-circuit-directed-graph/)
13. [Biconnected Components](https://www.geeksforgeeks.org/biconnected-components/)
14. [Tarjan’s Algorithm to find Strongly Connected Components](https://www.geeksforgeeks.org/tarjan-algorithm-find-strongly-connected-components/)

***Hard Problems:***

1. [Graph Coloring (Introduction and Applications)](https://www.geeksforgeeks.org/graph-coloring-applications/)
2. [Greedy Algorithm for Graph Coloring](https://www.geeksforgeeks.org/graph-coloring-set-2-greedy-algorithm/)
3. [Travelling Salesman Problem (Naive and Dynamic Programming)](https://www.geeksforgeeks.org/travelling-salesman-problem-set-1/)
4. [Travelling Salesman Problem (Approximate using MST)](https://www.geeksforgeeks.org/travelling-salesman-problem-set-2-approximate-using-mst/)
5. [Hamiltonian Cycle](https://www.geeksforgeeks.org/backtracking-set-7-hamiltonian-cycle/)
6. [Vertex Cover Problem (Introduction and Approximate Algorithm)](https://www.geeksforgeeks.org/vertex-cover-problem-set-1-introduction-approximate-algorithm-2/)
7. [K Centers Problem (Greedy Approximate Algorithm)](https://www.geeksforgeeks.org/k-centers-problem-set-1-greedy-approximate-algorithm/)

***Maximum Flow:***

1. [Ford-Fulkerson Algorithm for Maximum Flow Problem](https://www.geeksforgeeks.org/ford-fulkerson-algorithm-for-maximum-flow-problem/)
2. [Find maximum number of edge disjoint paths between two vertices](https://www.geeksforgeeks.org/find-edge-disjoint-paths-two-vertices/)
3. [Find minimum s-t cut in a flow network](https://www.geeksforgeeks.org/minimum-cut-in-a-directed-graph/)
4. [Maximum Bipartite Matching](https://www.geeksforgeeks.org/maximum-bipartite-matching/)
5. [Channel Assignment Problem](https://www.geeksforgeeks.org/channel-assignment-problem/)

**Misc:**

0. [Karger’s algorithm for Minimum Cut](https://www.geeksforgeeks.org/kargers-algorithm-for-minimum-cut-set-1-introduction-and-implementation/)

1. [Karger’s algorithm for Minimum Cut | Set 2 (Analysis and Applications)](https://www.geeksforgeeks.org/kargers-algorithm-for-minimum-cut-set-2-analysis-and-applications/" \o "Permalink to Karger’s algorithm for Minimum Cut | Set 2 (Analysis and Applications))
2. [Hopcroft–Karp Algorithm for Maximum Matching | Set 1 (Introduction)](https://www.geeksforgeeks.org/hopcroft-karp-algorithm-for-maximum-matching-set-1-introduction/)
3. [Hopcroft–Karp Algorithm for Maximum Matching | Set 2 (Implementation)](https://www.geeksforgeeks.org/hopcroft-karp-algorithm-for-maximum-matching-set-1-introduction/)

**Randomized Algorithms:**

1. [Linearity of Expectation](https://www.geeksforgeeks.org/linearity-of-expectation/)
2. [Expected Number of Trials until Success](https://www.geeksforgeeks.org/expected-number-of-trials-before-success/)
3. [Randomized Algorithms | Set 0 (Mathematical Background)](https://www.geeksforgeeks.org/randomized-algorithms-set-0-mathematical-background/)
4. [Randomized Algorithms | Set 1 (Introduction and Analysis)](https://www.geeksforgeeks.org/randomized-algorithms-set-1-introduction-and-analysis/)
5. [Randomized Algorithms | Set 2 (Classification and Applications)](https://www.geeksforgeeks.org/randomized-algorithms-set-2-classification-and-applications/)
6. [Randomized Algorithms | Set 3 (1/2 Approximate Median)](https://www.geeksforgeeks.org/randomized-algorithms-set-3-12-approximate-median/)
7. [Karger’s algorithm for Minimum Cut](https://www.geeksforgeeks.org/kargers-algorithm-for-minimum-cut-set-1-introduction-and-implementation/)
8. [K’th Smallest/Largest Element in Unsorted Array | Set 2 (Expected Linear Time)](https://www.geeksforgeeks.org/kth-smallestlargest-element-unsorted-array-set-2-expected-linear-time/)
9. [Reservoir Sampling](https://www.geeksforgeeks.org/reservoir-sampling/)
10. [Shuffle a given array](https://www.geeksforgeeks.org/shuffle-a-given-array/)
11. [Select a Random Node from a Singly Linked List](https://www.geeksforgeeks.org/select-a-random-node-from-a-singly-linked-list/)

**Branch and Bound**:

1. [Branch and Bound | Set 1 (Introduction with 0/1 Knapsack)](https://www.geeksforgeeks.org/branch-and-bound-set-1-introduction-with-01-knapsack/)
2. [Branch and Bound | Set 2 (Implementation of 0/1 Knapsack)](https://www.geeksforgeeks.org/branch-and-bound-set-2-implementation-of-01-knapsack/)
3. [Branch and Bound | Set 3 (8 puzzle Problem)](https://www.geeksforgeeks.org/branch-bound-set-3-8-puzzle-problem/)
4. [Branch And Bound | Set 4 (Job Assignment Problem)](https://www.geeksforgeeks.org/branch-bound-set-4-job-assignment-problem/)
5. [Branch and Bound | Set 5 (N Queen Problem)](https://www.geeksforgeeks.org/branch-and-bound-set-4-n-queen-problem/)
6. [Branch And Bound | Set 6 (Traveling Salesman Problem)](https://www.geeksforgeeks.org/branch-bound-set-5-traveling-salesman-problem/)