# **The ultimate reference to competitive coding**

# Topics :

**Stack :**

<https://www.geeksforgeeks.org/stack-data-structure/>

<https://practice.geeksforgeeks.org/explore/?category%5B%5D=Stack&page=1&category%5B%5D=Stack>

<https://leetcode.com/tag/stack/>

**Queue:**

<https://leetcode.com/tag/queue/>

<https://www.geeksforgeeks.org/queue-data-structure/>

**Linked List:**

<https://www.geeksforgeeks.org/data-structures/linked-list/>

<https://www.geeksforgeeks.org/linked-list-set-1-introduction/>

<https://leetcode.com/explore/learn/card/linked-list/>

**Binary Tree:**

<https://www.geeksforgeeks.org/binary-tree-data-structure/>

<https://leetcode.com/tag/tree/>

**Binary Search Tree:**

<https://leetcode.com/tag/binary-search-tree/>

<https://leetcode.com/explore/learn/card/data-structure-tree/>

<https://www.geeksforgeeks.org/binary-search-tree-set-1-search-and-insertion/>

**Graph:**

<https://www.geeksforgeeks.org/graph-data-structure-and-algorithms/>

<https://leetcode.com/tag/graph/>

**Pattern Printing :**

<https://www.geeksforgeeks.org/tag/pattern-printing/>

**Sliding Window :**

<https://leetcode.com/tag/sliding-window/>

**Prefix Sum :**

<https://www.codechef.com/tags/problems/prefix-sum>

**Hash Table :**

<https://leetcode.com/tag/hash-table/>

<https://www.codechef.com/tags/problems/hashing>

**Number Theory :**

<https://www.hackerearth.com/practice/math/number-theory/basic-number-theory-1/practice-problems/3/?sort_by=partially%20solved&p_level=>

<https://www.geeksforgeeks.org/number-theory-competitive-programming/>

<https://www.codechef.com/wiki/tutorial-number-theory>

**Linear Search :**

<https://www.hackerearth.com/practice/algorithms/searching/linear-search/practice-problems/>

**Binary Search :**

<https://leetcode.com/tag/binary-search/>

<https://www.interviewbit.com/courses/programming/topics/binary-search/>

<https://www.hackerearth.com/practice/algorithms/searching/binary-search/practice-problems/>

**Euclidean Algorithm( For GCD) :**

<https://www.geeksforgeeks.org/euclidean-algorithms-basic-and-extended/>

<https://www.hackerearth.com/problem/algorithm/gcd-17-8064d146-635f817b/>

**Linear Diophantine Equation :**

<https://www.geeksforgeeks.org/linear-diophantine-equations/>

**Primality Problems :**

<https://dev.to/priyanka__488/number-theory-primality-test-in-o-sqrt-n-dde>

**Sieve Of Erathenes:**

<https://www.geeksforgeeks.org/how-is-the-time-complexity-of-sieve-of-eratosthenes-is-nloglogn/>

<https://www.geeksforgeeks.org/sieve-of-eratosthenes/amp/>

<https://www.geeksforgeeks.org/segmented-sieve/>

<https://www.geeksforgeeks.org/sieve-eratosthenes-0n-time-complexity/>

<https://www.geeksforgeeks.org/prime-factorization-using-sieve-olog-n-multiple-queries/>

<https://www.codechef.com/tags/problems/prime-factorization>

**Eulers Totient Function:**

<https://www.geeksforgeeks.org/eulers-totient-function/>

**Fermats little Theorem:**

<https://www.geeksforgeeks.org/fermats-little-theorem/>

**Wilson’s Theorem:**

<https://www.geeksforgeeks.org/wilsons-theorem/>

**Calculate Power of x to n:**

<https://www.geeksforgeeks.org/write-a-c-program-to-calculate-powxn/>

**Modular Arithmetic:**

<https://www.hackersfriend.com/articles/modular-arithmetic-for-competitive-programming>

<https://www.geeksforgeeks.org/multiplicative-inverse-under-modulo-m/>

<https://cp-algorithms.com/algebra/module-inverse.html>

<https://www.geeksforgeeks.org/modular-exponentiation-power-in-modular-arithmetic/>

**Chinese Remainder Theorem:**

<https://www.geeksforgeeks.org/chinese-remainder-theorem-set-1-introduction/>

**Modulo problems:**

<https://www.geeksforgeeks.org/number-of-factors-of-very-large-number-n-modulo-m-where-m-is-any-prime-number/>

<https://www.geeksforgeeks.org/modulo-1097-1000000007/>

<https://www.geeksforgeeks.org/compute-n-under-modulo-p/>

Ncr **under constant time complexity:**

<https://www.geeksforgeeks.org/queries-of-ncrp-in-o1-time-complexity/>

**Inclustion – Exclusion applications:**

<https://www.geeksforgeeks.org/inclusion-exclusion-various-applications/amp/#aoh=16327936316328&amp_ct=1632793636730&referrer=https%3A%2F%2Fwww.google.com&amp_tf=From%20%251%24s>

**Sorting:**

<https://www.geeksforgeeks.org/sorting-algorithms/>

<https://www.codechef.com/tags/problems/sorting>

**Constructive algorithms:**

<https://www.geeksforgeeks.org/easy/constructive-algorithms/>

**Two Pointers:**

<https://leetcode.com/tag/two-pointers/>

<https://www.interviewbit.com/courses/programming/topics/two-pointers/>

<https://www.geeksforgeeks.org/two-pointers-technique/>

**Bit manipulation:**

<https://leetcode.com/tag/bit-manipulation/>

<https://www.interviewbit.com/courses/programming/topics/bit-manipulation/>

<https://www.geeksforgeeks.org/power-set/>

<https://www.geeksforgeeks.org/bitwise-algorithms/>

**Xor problems:**

<https://www.geeksforgeeks.org/tag/xor/>

<https://codeforces.com/blog/entry/68953>

<https://www.geeksforgeeks.org/count-number-subarrays-given-xor/>

<https://leetcode.com/problems/xor-queries-of-a-subarray/description/>

<https://codeforces.com/blog/entry/53782>

**Greedy Algorithms:**

<https://www.geeksforgeeks.org/greedy-algorithms/>

<https://practice.geeksforgeeks.org/explore/?category%5B%5D=Greedy&page=1&category%5B%5D=Greedy>

<https://cp-algorithms.com/>

<https://www.interviewbit.com/courses/programming/topics/greedy-algorithm/>

<https://leetcode.com/tag/greedy/>

**Maximum average segment:**

<https://cp-algorithms.com/others/maximum_average_segment.html>

<https://codeforces.com/blog/entry/13713>

**Prefix Sum:**

<https://www.geeksforgeeks.org/prefix-sum-array-implementation-applications-competitive-programming/>

**Array and Subarray sum problems:**

<https://www.geeksforgeeks.org/largest-sum-contiguous-subarray/>

<https://leetcode.com/problems/best-time-to-buy-and-sell-stock/discuss/39038/Kadane%27s-Algorithm-Since-no-one-has-mentioned-about-this-so-far-:%29-%28In-case-if-interviewer-twists-the-input%29>

<https://www.geeksforgeeks.org/tag/subarray/>

**Job sequencing && Activity Selection Problems:**

<https://www.geeksforgeeks.org/job-sequencing-problem/>

<https://www.geeksforgeeks.org/activity-selection-problem-greedy-algo-1/>

<https://www.interviewbit.com/tutorial/activity-selection-problem/>

**Recursion:**

<https://www.geeksforgeeks.org/recursion/>

<https://www.geeksforgeeks.org/recursion-practice-problems-solutions/>

<https://leetcode.com/tag/recursion/>

<https://leetcode.com/explore/learn/card/recursion-ii/>

<https://leetcode.com/explore/learn/card/recursion-i/>

<https://www.codechef.com/tags/problems/recursion>

<https://www.geeksforgeeks.org/recursive-program-to-print-all-subsets-with-given-sum/>

**Merge Sort :**

<https://www.geeksforgeeks.org/merge-sort/amp/>

<https://www.hackerearth.com/practice/algorithms/sorting/merge-sort/practice-problems/>

<https://www.geeksforgeeks.org/tag/merge-sort/>

<https://practice.geeksforgeeks.org/problems/merge-sort/1>

<https://leetcode.com/tag/merge-sort/>

**Quick Sort:**

<https://www.geeksforgeeks.org/quick-sort/>

**Backtraking:**

<https://www.geeksforgeeks.org/backtracking-introduction/>

<https://www.geeksforgeeks.org/backtracking-algorithms/>

<https://practice.geeksforgeeks.org/topics/backtracking/>

**Meet in the middle Problems:**

<https://www.geeksforgeeks.org/meet-in-the-middle/>

<https://www.codechef.com/tags/problems/meet-in-middle>

<https://codeforces.com/problemset?order=BY_RATING_ASC&tags=meet-in-the-middle>

**Divide and Conquer Problems:**

<https://www.geeksforgeeks.org/divide-and-conquer-algorithm-introduction/>

<https://www.geeksforgeeks.org/divide-and-conquer/>

<https://leetcode.com/tag/divide-and-conquer/>

<https://www.codechef.com/tags/problems/divide-and-conquer>

**Next greater and smaller element:**

<https://www.geeksforgeeks.org/next-greater-element/>

<https://www.geeksforgeeks.org/next-smaller-element/>

**Paranthesis Problems:**

<https://www.geeksforgeeks.org/tag/parentheses-problems/>

**Largest Rectangular Area in a histogram:**

<https://www.geeksforgeeks.org/largest-rectangular-area-in-a-histogram-set-1/>

<https://www.geeksforgeeks.org/largest-rectangle-under-histogram/>

**Heap Data Structure:**

<https://www.geeksforgeeks.org/heap-data-structure/>

<https://leetcode.com/tag/heap-priority-queue/>

<https://leetcode.com/discuss/general-discussion/1127238/master-heap-by-solving-23-questions-in-4-patterns-category>

<https://leetcode.com/problems/merge-k-sorted-lists/discuss/10527/Difference-between-Priority-Queue-and-Heap-and-C%2B%2B-implementation>

<https://www.geeksforgeeks.org/heap-using-stl-c/>

<https://www.hackerearth.com/practice/data-structures/trees/heapspriority-queues/practice-problems/>

**String hashing:**

<https://www.geeksforgeeks.org/string-hashing-using-polynomial-rolling-hash-function/>

<https://discuss.codechef.com/t/classical-string-hashing-problem/19288>

**Rabin Karp algorithm:**

<https://www.geeksforgeeks.org/rabin-karp-algorithm-for-pattern-searching/>

<https://practice.geeksforgeeks.org/problems/31272eef104840f7430ad9fd1d43b434a4b9596b/1/>

<https://cp-algorithms.com/string/prefix-function.html>

**Kmp algorithm:**

<https://www.geeksforgeeks.org/kmp-algorithm-for-pattern-searching/>

<https://practice.geeksforgeeks.org/problems/search-pattern0205/1>

**Z function:**

<https://cp-algorithms.com/string/z-function.html>

<https://www.geeksforgeeks.org/z-algorithm-linear-time-pattern-searching-algorithm/>

**Manacher Algorithm:**

<https://cp-algorithms.com/string/manacher.html>

<https://www.geeksforgeeks.org/manachers-algorithm-linear-time-longest-palindromic-substring-part-1/>

**Once you have wrapped all these algorithms,**

**problems on various platforms**

Graph problems:

<https://leetcode.com/tag/graph/>

**Diameter of binary Tree:**

<https://www.geeksforgeeks.org/diameter-of-a-binary-tree/>

**Euler Path :**

<https://cp-algorithms.com/graph/euler_path.html>

<https://www.geeksforgeeks.org/euler-tour-tree/>

<https://cp-algorithms.com/graph/lca.html>

**Find lca using Euler tour**

<https://cp-algorithms.com/graph/lca_binary_lifting.html>

<https://www.geeksforgeeks.org/lca-in-a-tree-using-binary-lifting-technique/>

<https://codeforces.com/blog/entry/20935>

**SubTree Sum Problem :**

<https://geekstocode.com/sub-tree-sum-problem-in-tree-data-structure/>

**Distance Between Nodes in a binary tree:**

<https://www.geeksforgeeks.org/find-distance-between-two-nodes-of-a-binary-tree/>

<https://practice.geeksforgeeks.org/problems/min-distance-between-two-given-nodes-of-a-binary-tree/1>

**Solve problems on above algorithms**

**Connected Components in undirected graph:**

<https://www.geeksforgeeks.org/connected-components-in-an-undirected-graph/>

**Topological Sorting:**

<https://www.geeksforgeeks.org/topological-sorting/>

**Detection of cycle in a graph:**

<https://www.geeksforgeeks.org/detect-cycle-in-a-graph/>

<https://www.geeksforgeeks.org/detect-cycle-undirected-graph/>

**Bipartite Graph:**

<https://www.geeksforgeeks.org/bipartite-graph/>

**Strongly connected components in graph:**

<https://cp-algorithms.com/graph/strongly-connected-components.html>

<https://www.geeksforgeeks.org/strongly-connected-components/>

**Dijkstra’s algorithm:**

<https://www.geeksforgeeks.org/dijkstras-shortest-path-algorithm-greedy-algo-7/>

**Bellman-ford-algorithm:**

<https://www.geeksforgeeks.org/bellman-ford-algorithm-dp-23/>

<https://cp-algorithms.com/graph/bellman_ford.html>

<https://www.techiedelight.com/single-source-shortest-paths-bellman-ford-algorithm/>

**Floyd-warshall-algorithm:**

<https://www.geeksforgeeks.org/floyd-warshall-algorithm-dp-16/>

<https://cp-algorithms.com/graph/all-pair-shortest-path-floyd-warshall.html>

<https://www.techiedelight.com/pairs-shortest-paths-floyd-warshall-algorithm/>

**Bridge in graph:**

<https://cp-algorithms.com/graph/bridge-searching.html>

<https://www.geeksforgeeks.org/bridge-in-a-graph/>

**Articulation points in graph:**

<https://www.geeksforgeeks.org/articulation-points-or-cut-vertices-in-a-graph/>

<https://www.hackerearth.com/practice/algorithms/graphs/articulation-points-and-bridges/tutorial/>

<https://cp-algorithms.com/graph/cutpoints.html>

**Kruskal’s Algorithm:**

<https://www.geeksforgeeks.org/kruskals-minimum-spanning-tree-algorithm-greedy-algo-2/>

<https://cp-algorithms.com/graph/mst_kruskal.html>

<https://www.geeksforgeeks.org/prims-minimum-spanning-tree-mst-greedy-algo-5/>

**Prim’s algorithm:**

<https://cp-algorithms.com/graph/mst_prim.html>

**Shortest path and BFS application:**

<https://www.geeksforgeeks.org/0-1-bfs-shortest-path-binary-graph/amp/>

<https://cp-algorithms.com/graph/01_bfs.html>

**Bridge searching online:**

<https://cp-algorithms.com/graph/bridge-searching-online.html>

**Solve problems on above**

**Digit DP:**

<https://www.geeksforgeeks.org/digit-dp-introduction/>

<https://codeforces.com/blog/entry/53960>

**Solve a bunch of problems on it**

**Bit masking:**

<https://www.geeksforgeeks.org/bitmasking-and-dynamic-programming-set-1-count-ways-to-assign-unique-cap-to-every-person/amp/>

<https://www.hackerearth.com/practice/algorithms/dynamic-programming/bit-masking/tutorial/>

<https://codeforces.com/blog/entry/13293>

<https://codeforces.com/blog/entry/81516>

<https://www.geeksforgeeks.org/bitmasking-and-dynamic-programming-set-1-count-ways-to-assign-unique-cap-to-every-person/>

<https://codeforces.com/blog/entry/20935>

**Dynamic Programming:**

<https://www.geeksforgeeks.org/dynamic-programming-trees-set-1/>

<https://www.geeksforgeeks.org/dynamic-programming-trees-set-2/>

**Dp on trees:**

<https://discuss.codechef.com/t/dp-on-trees-lecture-series-tutorial/16840>

<https://codeforces.com/blog/entry/20935>

<https://codeforces.com/blog/entry/45223>

<https://youtu.be/mkiK_GCWX50>

<https://www.geeksforgeeks.org/sum-subsets-dynamic-programming/amp/>

**Solve as many as you can**

**Disjoint Set Union:**

<https://cp-algorithms.com/data_structures/disjoint_set_union.html>

<https://codeforces.com/blog/entry/75066>

<https://www.geeksforgeeks.org/union-find-algorithm-set-2-union-by-rank/>

<https://www.hackerearth.com/practice/notes/disjoint-set-union-union-find/>

**Sparse Table:**

<https://cp-algorithms.com/data_structures/sparse-table.html>

<https://www.geeksforgeeks.org/sparse-table/>

<https://www.hackerearth.com/practice/notes/sparse-table/>

<https://codeforces.com/blog/entry/66643>

**Fenwick Tree (aka Binary Indexed Tree):**

<https://cp-algorithms.com/data_structures/fenwick.html>

<https://www.geeksforgeeks.org/binary-indexed-tree-or-fenwick-tree-2/>

<https://www.hackerearth.com/practice/notes/binary-indexed-tree-or-fenwick-tree/>

<https://leetcode.com/problems/finding-mk-average/discuss/1157928/binary-indexed-trees-bit-or-fenwick-tree-binary-lifting>

**Solve problems on Fenwick tree**

**Matrix exponentiation**

<https://www.geeksforgeeks.org/matrix-exponentiation/>

<https://www.hackerearth.com/practice/notes/matrix-exponentiation-1/>

<https://codeforces.com/blog/entry/67776>

<https://codeforces.com/blog/entry/8544>

**Sqrt Decomposition:**

<https://cp-algorithms.com/data_structures/sqrt_decomposition.html>

<https://www.geeksforgeeks.org/sqrt-square-root-decomposition-technique-set-1-introduction/>

<https://discuss.codechef.com/t/range-update-in-square-root-decomposition-technique/13181>

<https://codeforces.com/blog/entry/23005>

<https://codeforces.com/blog/entry/83248>

<https://cp-algorithms.com/data_structures/sqrt_decomposition.html>

<https://www.geeksforgeeks.org/mos-algorithm-query-square-root-decomposition-set-1-introduction/>

<https://codeforces.com/blog/entry/43230>

**Mo’s Algorithm:**

<https://www.geeksforgeeks.org/mos-algorithm-query-square-root-decomposition-set-1-introduction/>

<https://www.hackerearth.com/practice/notes/mos-algorithm/>

**Segment Trees:**

<https://cp-algorithms.com/data_structures/segment_tree.html>

<https://www.hackerearth.com/practice/data-structures/advanced-data-structures/segment-trees/tutorial/>

<https://www.geeksforgeeks.org/segment-tree-set-1-sum-of-given-range/amp/>

<https://www.geeksforgeeks.org/lazy-propagation-in-segment-tree/>

<https://www.hackerearth.com/practice/notes/segment-tree-and-lazy-propagation/>

<https://www.geeksforgeeks.org/segment-tree-set-2-range-maximum-query-node-update/>

**Spargue Grundy Theorm:**

<https://www.geeksforgeeks.org/combinatorial-game-theory-set-4-sprague-grundy-theorem/>

<https://cp-algorithms.com/game_theory/sprague-grundy-nim.html>

**Flows and related problems:**

<https://www.geeksforgeeks.org/max-flow-problem-introduction/>

<https://www.hackerearth.com/practice/algorithms/graphs/maximum-flow/tutorial/>

<https://cp-algorithms.com/graph/flow_with_demands.html>

<https://cp-algorithms.com/graph/edmonds_karp.html>

<https://cp-algorithms.com/graph/dinic.html>

<https://cp-algorithms.com/graph/min_cost_flow.html>

**Heavy light decomposition:**

<https://cp-algorithms.com/graph/hld.html>

<https://www.geeksforgeeks.org/heavy-light-decomposition-set-1-introduction/>

**Convex hull theorem:**

<https://www.geeksforgeeks.org/convex-hull-set-1-jarviss-algorithm-or-wrapping/>

<https://cp-algorithms.com/geometry/grahams-scan-convex-hull.html>

**FFT/NTT:**

<https://cp-algorithms.com/algebra/fft.html>

<https://www.geeksforgeeks.org/fast-fourier-transformation-poynomial-multiplication/>

<https://gist.github.com/jakobkogler/c320233e9275f3b6badc12988e9e9e47>

<https://codeforces.com/blog/entry/43499>

**O(n) approach for checking binary string divisible by 5 or not**

<https://www.geeksforgeeks.org/decimal-representation-given-binary-string-divisible-5-not/>

For all the algorithm , refer to the CP algorithm website