Algorithms

**Topics:**

* [Analysis of Algorithms](https://www.geeksforgeeks.org/fundamentals-of-algorithms/#AnalysisofAlgorithms)
* [Searching and Sorting](https://www.geeksforgeeks.org/fundamentals-of-algorithms/#SearchingandSorting)
* [Greedy Algorithms](https://www.geeksforgeeks.org/fundamentals-of-algorithms/#GreedyAlgorithms)
* [Dynamic Programming](https://www.geeksforgeeks.org/fundamentals-of-algorithms/#DynamicProgramming)
* [Pattern Searching](https://www.geeksforgeeks.org/fundamentals-of-algorithms/#PatternSearching)
* [Other String Algorithms](https://www.geeksforgeeks.org/fundamentals-of-algorithms/#OtherStringAlgorithms)
* [Backtracking](https://www.geeksforgeeks.org/fundamentals-of-algorithms/#Backtracking)
* [Divide and Conquer](https://www.geeksforgeeks.org/fundamentals-of-algorithms/#DivideandConquer)
* [Geometric Algorithms](https://www.geeksforgeeks.org/fundamentals-of-algorithms/#GeometricAlgorithms)
* [Mathematical Algorithms](https://www.geeksforgeeks.org/fundamentals-of-algorithms/#MathematicalAlgorithms)
* [Bit Algorithms](https://www.geeksforgeeks.org/fundamentals-of-algorithms/#BitAlgorithms)
* [Graph Algorithms](https://www.geeksforgeeks.org/fundamentals-of-algorithms/#GraphAlgorithms)
* [Randomized Algorithms](https://www.geeksforgeeks.org/fundamentals-of-algorithms/#RandomizedAlgorithms)
* [Branch and Bound](https://www.geeksforgeeks.org/fundamentals-of-algorithms/#BranchandBound)
* [Quizzes on Algorithms](https://www.geeksforgeeks.org/fundamentals-of-algorithms/#QuizzesonAlgorithms)
* [Misc](https://www.geeksforgeeks.org/fundamentals-of-algorithms/#Misc)

**Analysis of Algorithms**:

1. [Asymptotic Analysis](https://www.geeksforgeeks.org/analysis-of-algorithms-set-1-asymptotic-analysis/)
2. [Worst, Average and Best Cases](https://www.geeksforgeeks.org/analysis-of-algorithms-set-2-asymptotic-analysis/)
3. [Asymptotic Notations](https://www.geeksforgeeks.org/analysis-of-algorithms-set-3asymptotic-notations/)
4. [Little o and little omega notations](https://www.geeksforgeeks.org/analysis-of-algorithems-little-o-and-little-omega-notations/)
5. [Lower and Upper Bound Theory](https://www.geeksforgeeks.org/lower-and-upper-bound-theory/)
6. [Analysis of Loops](https://www.geeksforgeeks.org/analysis-algorithms-set-4-analysis-programs/)
7. [Solving Recurrences](https://www.geeksforgeeks.org/analysis-algorithm-set-4-master-method-solving-recurrences/)
8. [Amortized Analysis](https://www.geeksforgeeks.org/analysis-algorithm-set-5-amortized-analysis-introduction/)
9. [What does ‘Space Complexity’ mean?](https://www.geeksforgeeks.org/g-fact-86/)
10. [Pseudo-polynomial Algorithms](https://www.geeksforgeeks.org/pseudo-polynomial-in-algorithms/)
11. [NP-Completeness Introduction](https://www.geeksforgeeks.org/np-completeness-set-1/)
12. [Polynomial Time Approximation Scheme](https://www.geeksforgeeks.org/polynomial-time-approximation-scheme/)
13. [A Time Complexity Question](https://www.geeksforgeeks.org/a-time-complexity-question/)
14. [Time Complexity of building a heap](https://www.geeksforgeeks.org/g-fact-85/)
15. [Time Complexity where loop variable is incremented by 1, 2, 3, 4 ..](https://www.geeksforgeeks.org/time-complexity-where-loop-variable-is-incremented-by-1-2-3-4/)
16. [Time Complexity of Loop with Powers](https://www.geeksforgeeks.org/time-complexity-of-loop-with-powers/)
17. [Performance of loops (A caching question)](http://geeksquiz.com/performance-of-loops-a-caching-question/)

[**Recent Articles on Analysis of Algorithms**](https://www.geeksforgeeks.org/category/algorithm/analysis/)  
[Quiz on Analysis of Algorithms](http://geeksquiz.com/algorithms/analysis-of-algorithms/)  
[Quiz on Recurrences](http://geeksquiz.com/algorithms/analysis-of-algorithms-recurrences/)

**Searching and Sorting**:

1. [Linear Search](http://quiz.geeksforgeeks.org/linear-search/), [Binary Search](http://geeksquiz.com/binary-search/), [Jump Search](https://www.geeksforgeeks.org/jump-search/), [Interpolation Search](https://www.geeksforgeeks.org/interpolation-search/), [Exponential Search](https://www.geeksforgeeks.org/exponential-search/), [Ternary Search](https://www.geeksforgeeks.org/binary-search-preferred-ternary-search/)
2. [Selection Sort](http://geeksquiz.com/selection-sort/), [Bubble Sort](http://geeksquiz.com/bubble-sort/), [Insertion Sort](http://geeksquiz.com/insertion-sort/), [Merge Sort](http://geeksquiz.com/merge-sort/), [Heap Sort](http://geeksquiz.com/heap-sort/), [Quicksort](http://geeksquiz.com/quick-sort/), [Radix Sort](https://www.geeksforgeeks.org/radix-sort/), [Counting Sort](https://www.geeksforgeeks.org/counting-sort/), [Bucket Sort](https://www.geeksforgeeks.org/bucket-sort-2/), [ShellSort](http://geeksquiz.com/shellsort/), [Comb Sort](https://www.geeksforgeeks.org/comb-sort/), [Pigeonhole Sort](https://www.geeksforgeeks.org/pigeonhole-sort/), [Cycle Sort](https://www.geeksforgeeks.org/cycle-sort/)
3. [Interpolation search vs Binary search](https://www.geeksforgeeks.org/g-fact-84/)
4. [Stability in sorting algorithms](https://www.geeksforgeeks.org/stability-in-sorting-algorithms/)
5. [When does the worst case of Quicksort occur?](https://www.geeksforgeeks.org/when-does-the-worst-case-of-quicksort-occur/)
6. [Lower bound for comparison based sorting algorithms](https://www.geeksforgeeks.org/lower-bound-on-comparison-based-sorting-algorithms/)
7. [Which sorting algorithm makes minimum number of memory writes?](https://www.geeksforgeeks.org/which-sorting-algorithm-makes-minimum-number-of-writes/)
8. [Find the Minimum length Unsorted Subarray, sorting which makes the complete array sorted](https://www.geeksforgeeks.org/minimum-length-unsorted-subarray-sorting-which-makes-the-complete-array-sorted/)
9. [Merge Sort for Linked Lists](https://www.geeksforgeeks.org/merge-sort-for-linked-list/)
10. [Sort a nearly sorted (or K sorted) array](https://www.geeksforgeeks.org/nearly-sorted-algorithm/)
11. [Iterative Quick Sort](https://www.geeksforgeeks.org/iterative-quick-sort/)
12. [QuickSort on Singly Linked List](https://www.geeksforgeeks.org/quicksort-on-singly-linked-list/)
13. [QuickSort on Doubly Linked List](https://www.geeksforgeeks.org/quicksort-for-linked-list/)
14. [Find k closest elements to a given value](https://www.geeksforgeeks.org/find-k-closest-elements-given-value/)
15. [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/)
16. [A Problem in Many Binary Search Implementations](https://www.geeksforgeeks.org/problem-binary-search-implementations/)
17. [Search in an almost sorted array](https://www.geeksforgeeks.org/search-almost-sorted-array/)
18. [Sort an array in wave form](https://www.geeksforgeeks.org/sort-array-wave-form-2/)
19. [Why is Binary Search preferred over Ternary Search?](https://www.geeksforgeeks.org/binary-search-preferred-ternary-search/)
20. [K’th Smallest/Largest Element in Unsorted Array](https://www.geeksforgeeks.org/kth-smallestlargest-element-unsorted-array/)
21. [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/)
22. [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/)
23. [Find the closest pair from two sorted arrays](https://www.geeksforgeeks.org/given-two-sorted-arrays-number-x-find-pair-whose-sum-closest-x/)
24. [Find common elements in three sorted arrays](https://www.geeksforgeeks.org/find-common-elements-three-sorted-arrays/)
25. [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/)
26. [Count 1’s in a sorted binary array](http://geeksquiz.com/count-1s-sorted-binary-array/)
27. [Binary Insertion Sort](http://geeksquiz.com/binary-insertion-sort/)
28. [Insertion Sort for Singly Linked List](http://geeksquiz.com/insertion-sort-for-singly-linked-list/)
29. [Why Quick Sort preferred for Arrays and Merge Sort for Linked Lists?](https://www.geeksforgeeks.org/why-quick-sort-preferred-for-arrays-and-merge-sort-for-linked-lists/)
30. [Merge Sort for Doubly Linked List](https://www.geeksforgeeks.org/merge-sort-for-doubly-linked-list/)
31. [Minimum adjacent swaps to move maximum and minimum to corners](https://www.geeksforgeeks.org/minimum-adjacent-swaps-to-move-maximum-and-minimum-to-corners/)

**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. [Quiz on Greedy Algorithms](http://geeksquiz.com/algorithms/greedy-algorithms/)
11. [Greedy Algorithm to find Minimum number of Coins](http://geeksquiz.com/greedy-algorithm-to-find-minimum-number-of-coins/)
12. [K Centers Problem](https://www.geeksforgeeks.org/k-centers-problem-set-1-greedy-approximate-algorithm/)
13. [Minimum Number of Platforms Required for a Railway/Bus Station](https://www.geeksforgeeks.org/minimum-number-platforms-required-railwaybus-station/)

**Dynamic Programming**:

1. [Overlapping Sub problems 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/)
30. [Optimal Binary Search Tree](https://www.geeksforgeeks.org/dynamic-programming-set-24-optimal-binary-search-tree/)
31. [Largest Independent Set Problem](https://www.geeksforgeeks.org/largest-independent-set-problem/)
32. [Subset Sum Problem](https://www.geeksforgeeks.org/dynamic-programming-subset-sum-problem/)
33. [Maximum sum rectangle in a 2D matrix](https://www.geeksforgeeks.org/dynamic-programming-set-27-max-sum-rectangle-in-a-2d-matrix/)
34. [Count number of binary strings without consecutive 1?s](https://www.geeksforgeeks.org/count-number-binary-strings-without-consecutive-1s/)
35. [Boolean Parenthesization Problem](https://www.geeksforgeeks.org/dynamic-programming-set-37-boolean-parenthesization-problem/)
36. [Count ways to reach the n’th stair](https://www.geeksforgeeks.org/count-ways-reach-nth-stair/)
37. [Minimum Cost Polygon Triangulation](https://www.geeksforgeeks.org/minimum-cost-polygon-triangulation/)
38. [Mobile Numeric Keypad Problem](https://www.geeksforgeeks.org/mobile-numeric-keypad-problem/)
39. [Count of n digit numbers whose sum of digits equals to given sum](https://www.geeksforgeeks.org/count-of-n-digit-numbers-whose-sum-of-digits-equals-to-given-sum/)
40. [Minimum Initial Points to Reach Destination](https://www.geeksforgeeks.org/minimum-positive-points-to-reach-destination/)
41. [Total number of non-decreasing numbers with n digits](https://www.geeksforgeeks.org/total-number-of-non-decreasing-numbers-with-n-digits/)
42. [Find length of the longest consecutive path from a given starting character](https://www.geeksforgeeks.org/find-length-of-the-longest-consecutive-path-in-a-character-matrix/)
43. [Tiling Problem](https://www.geeksforgeeks.org/tiling-problem/)
44. [Minimum number of squares whose sum equals to given number n](https://www.geeksforgeeks.org/minimum-number-of-squares-whose-sum-equals-to-given-number-n/)
45. [Find minimum number of coins that make a given value](https://www.geeksforgeeks.org/find-minimum-number-of-coins-that-make-a-change/)
46. [Collect maximum points in a grid using two traversals](https://www.geeksforgeeks.org/collect-maximum-points-in-a-grid-using-two-traversals/)
47. [Shortest Common Supersequence](https://www.geeksforgeeks.org/shortest-common-supersequence/)
48. [Compute sum of digits in all numbers from 1 to n](https://www.geeksforgeeks.org/count-sum-of-digits-in-numbers-from-1-to-n/)
49. [Count possible ways to construct buildings](https://www.geeksforgeeks.org/count-possible-ways-to-construct-buildings/)
50. [Maximum profit by buying and selling a share at most twice](https://www.geeksforgeeks.org/maximum-profit-by-buying-and-selling-a-share-at-most-twice/)
51. [How to print maximum number of A’s using given four keys](https://www.geeksforgeeks.org/how-to-print-maximum-number-of-a-using-given-four-keys/)
52. [Find the minimum cost to reach destination using a train](https://www.geeksforgeeks.org/find-the-minimum-cost-to-reach-a-destination-where-every-station-is-connected-in-one-direction/)
53. [Vertex Cover Problem | Set 2 (Dynamic Programming Solution for Tree)](https://www.geeksforgeeks.org/vertex-cover-problem-set-2-dynamic-programming-solution-tree/)
54. [Count number of ways to reach a given score in a game](https://www.geeksforgeeks.org/count-number-ways-reach-given-score-game/)
55. [Weighted Job Scheduling](https://www.geeksforgeeks.org/weighted-job-scheduling/)
56. [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/)

**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/)
14. [Program to wish Women’s Day](https://www.geeksforgeeks.org/program-wish-womens-day/)

**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 Colouring 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 Crypt arithmetic Puzzles](https://www.geeksforgeeks.org/backtracking-set-8-solving-cryptarithmetic-puzzles/)

**Divide and Conquer**:

1. [Introduction](https://www.geeksforgeeks.org/divide-and-conquer-set-1-find-closest-pair-of-points/)
2. [Write your own pow(x, n) to calculate x\*n](https://www.geeksforgeeks.org/write-a-c-program-to-calculate-powxn/)
3. [Median of two sorted arrays](https://www.geeksforgeeks.org/median-of-two-sorted-arrays/)
4. [Count Inversions](https://www.geeksforgeeks.org/counting-inversions/)
5. [Closest Pair of Points](https://www.geeksforgeeks.org/closest-pair-of-points/)
6. [Strassen’s Matrix Multiplication](https://www.geeksforgeeks.org/strassens-matrix-multiplication/)
7. [Quick Sort vs Merge Sort](https://www.geeksforgeeks.org/quick-sort-vs-merge-sort/)

**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. [Multiply two integers without using multiplication, division and bitwise operators, and no loops](https://www.geeksforgeeks.org/multiply-two-numbers-without-using-multiply-division-bitwise-operators-and-no-loops/)
8. [Print all combinations of points that can compose a given number](https://www.geeksforgeeks.org/print-all-combinations-of-points-that-can-compose-a-given-number/)
9. [Write you own Power without using multiplication(\*) and division(/) operators](https://www.geeksforgeeks.org/write-you-own-power-without-using-multiplication-and-division/)
10. [Program for Fibonacci numbers](https://www.geeksforgeeks.org/program-for-nth-fibonacci-number/)
11. [Average of a stream of numbers](https://www.geeksforgeeks.org/average-of-a-stream-of-numbers/)
12. [Count numbers that don’t contain 3](https://www.geeksforgeeks.org/count-numbers-that-dont-contain-3/)
13. [Magic Square](https://www.geeksforgeeks.org/magic-square/)
14. [Sieve of Eratosthenes](https://www.geeksforgeeks.org/sieve-of-eratosthenes/)
15. [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/)
16. [Find day of the week for a given date](https://www.geeksforgeeks.org/find-day-of-the-week-for-a-given-date/)
17. [DFA based division](https://www.geeksforgeeks.org/dfa-based-division/)
18. [Generate integer from 1 to 7 with equal probability](https://www.geeksforgeeks.org/generate-integer-from-1-to-7-with-equal-probability/)
19. [Given a number, find the next smallest palindrome](https://www.geeksforgeeks.org/given-a-number-find-next-smallest-palindrome-larger-than-this-number/)
20. [Make a fair coin from a biased coin](https://www.geeksforgeeks.org/print-0-and-1-with-50-probability/)
21. [Check divisibility by 7](https://www.geeksforgeeks.org/divisibility-by-7/)
22. [Find the largest multiple of 3](https://www.geeksforgeeks.org/find-the-largest-number-multiple-of-3/)
23. [Lexicographic rank of a string](https://www.geeksforgeeks.org/lexicographic-rank-of-a-string/)
24. [Print all permutations in sorted (lexicographic) order](https://www.geeksforgeeks.org/lexicographic-permutations-of-string/)
25. [Shuffle a given array](https://www.geeksforgeeks.org/shuffle-a-given-array/)
26. [Space and time efficient Binomial Coefficient](https://www.geeksforgeeks.org/space-and-time-efficient-binomial-coefficient/)
27. [Reservoir Sampling](https://www.geeksforgeeks.org/reservoir-sampling/)
28. [Pascal’s Triangle](https://www.geeksforgeeks.org/pascal-triangle/)
29. [Select a random number from stream, with O(1) space](https://www.geeksforgeeks.org/select-a-random-number-from-stream-with-o1-space/)
30. [Find the largest multiple of 2, 3 and 5](https://www.geeksforgeeks.org/find-the-largest-multiple-of-2-3-and-5/)
31. [Efficient program to calculate e^x](https://www.geeksforgeeks.org/program-to-efficiently-calculate-ex/)
32. [Measure one litre using two vessels and infinite water supply](https://www.geeksforgeeks.org/measure-1-litre-from-two-vessels-infinite-water-supply/)
33. [Efficient program to print all prime factors of a given number](https://www.geeksforgeeks.org/print-all-prime-factors-of-a-given-number/)
34. [Print all possible combinations of r elements in a given array of size n](https://www.geeksforgeeks.org/print-all-possible-combinations-of-r-elements-in-a-given-array-of-size-n/)
35. [Random number generator in arbitrary probability distribution fashion](https://www.geeksforgeeks.org/random-number-generator-in-arbitrary-probability-distribution-fashion/)
36. [How to check if a given number is Fibonacci number?](https://www.geeksforgeeks.org/check-number-fibonacci-number/)
37. [Russian Peasant Multiplication](https://www.geeksforgeeks.org/fast-multiplication-method-without-using-multiplication-operator-russian-peasants-algorithm/)
38. [Count all possible groups of size 2 or 3 that have sum as multiple of 3](https://www.geeksforgeeks.org/count-possible-groups-size-2-3-sum-multiple-3/)
39. [Tower of Hanoi](http://geeksquiz.com/c-program-for-tower-of-hanoi/)
40. [Horner’s Method for Polynomial Evaluation](https://www.geeksforgeeks.org/horners-method-polynomial-evaluation/)
41. [Count trailing zeroes in factorial of a number](https://www.geeksforgeeks.org/count-trailing-zeroes-factorial-number/)
42. [Program for nth Catalan Number](https://www.geeksforgeeks.org/program-nth-catalan-number/)
43. [Generate one of 3 numbers according to given probabilities](https://www.geeksforgeeks.org/write-a-function-to-generate-3-numbers-according-to-given-probabilities/)
44. [Find Excel column name from a given column number](https://www.geeksforgeeks.org/find-excel-column-name-given-number/)
45. [Find next greater number with same set of digits](https://www.geeksforgeeks.org/find-next-greater-number-set-digits/)
46. [Count Possible Decodings of a given Digit Sequence](https://www.geeksforgeeks.org/count-possible-decodings-given-digit-sequence/)
47. [Calculate the angle between hour hand and minute hand](https://www.geeksforgeeks.org/calculate-angle-hour-hand-minute-hand/)
48. [Count number of binary strings without consecutive 1?s](https://www.geeksforgeeks.org/count-number-binary-strings-without-consecutive-1s/)
49. [Find the smallest number whose digits multiply to a given number n](https://www.geeksforgeeks.org/find-smallest-number-whose-digits-multiply-given-number-n/)
50. [Draw a circle without floating point arithmetic](http://geeksquiz.com/draw-circle-without-floating-point-arithmetic/)
51. [How to check if an instance of 8 puzzle is solvable?](https://www.geeksforgeeks.org/check-instance-8-puzzle-solvable/)
52. [Birthday Paradox](https://www.geeksforgeeks.org/birthday-paradox/)
53. [Multiply two polynomials](https://www.geeksforgeeks.org/multiply-two-polynomials-2/)
54. [Count Distinct Non-Negative Integer Pairs (x, y) that Satisfy the Inequality x\*x + y\*y < n](https://www.geeksforgeeks.org/count-distinct-non-negative-pairs-x-y-satisfy-inequality-xx-yy-n-2/)
55. [Count ways to reach the n’th stair](https://www.geeksforgeeks.org/count-ways-reach-nth-stair/)
56. [Replace all ‘0’ with ‘5’ in an input Integer](http://geeksquiz.com/replace-0-5-input-integer/)
57. [Program to add two polynomials](http://geeksquiz.com/program-add-two-polynomials/)
58. [Print first k digits of 1/n where n is a positive integer](http://geeksquiz.com/print-first-k-digits-1n-n-positive-integer/)
59. [Given a number as a string, find the number of contiguous subsequences which recursively add up to 9](http://geeksquiz.com/given-number-find-number-contiguous-subsequences-recursively-add-9/)
60. [Program for Bisection Method](https://www.geeksforgeeks.org/program-for-bisection-method/)
61. [Program for Method Of False Position](https://www.geeksforgeeks.org/program-for-method-of-false-position/)
62. [Program for Newton Raphson Method](https://www.geeksforgeeks.org/program-for-newton-raphson-method/)

**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. [Add 1 to a number](https://www.geeksforgeeks.org/add-1-to-a-given-number/)
12. [Multiply with 3.5](https://www.geeksforgeeks.org/multiply-an-integer-with-3-5/)
13. [Turn off the rightmost set bit](https://www.geeksforgeeks.org/turn-off-the-rightmost-set-bit/)
14. [Check for Power of 4](https://www.geeksforgeeks.org/find-whether-a-given-number-is-a-power-of-4-or-not/)
15. [Absolute value (abs) without branching](https://www.geeksforgeeks.org/compute-the-integer-absolute-value-abs-without-branching/)
16. [Modulus division by a power-of-2-number](https://www.geeksforgeeks.org/compute-modulus-division-by-a-power-of-2-number/)
17. [Minimum or Maximum of two integers](https://www.geeksforgeeks.org/compute-the-minimum-or-maximum-max-of-two-integers-without-branching/)
18. [Rotate bits](https://www.geeksforgeeks.org/rotate-bits-of-an-integer/)
19. [Find the two non-repeating elements in an array](https://www.geeksforgeeks.org/find-two-non-repeating-elements-in-an-array-of-repeating-elements/)
20. [Number Occurring Odd Number of Times](https://www.geeksforgeeks.org/find-the-number-occurring-odd-number-of-times/)
21. [Check for Integer Overflow](https://www.geeksforgeeks.org/check-for-integer-overflow/)
22. [Little and Big Endian](https://www.geeksforgeeks.org/little-and-big-endian-mystery/)
23. [Reverse Bits of a Number](https://www.geeksforgeeks.org/write-an-efficient-c-program-to-reverse-bits-of-a-number/)
24. [Count set bits in an integer](https://www.geeksforgeeks.org/count-set-bits-in-an-integer/)
25. [Number of bits to be flipped to convert A to B](https://www.geeksforgeeks.org/count-number-of-bits-to-be-flipped-to-convert-a-to-b/)
26. [Next Power of 2](https://www.geeksforgeeks.org/next-power-of-2/)
27. [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/)
28. [Find parity](https://www.geeksforgeeks.org/write-a-c-program-to-find-the-parity-of-an-unsigned-integer/)
29. [Multiply with 7](https://www.geeksforgeeks.org/efficient-way-to-multiply-with-7/)
30. [Find whether a no is power of two](https://www.geeksforgeeks.org/write-one-line-c-function-to-find-whether-a-no-is-power-of-two/)
31. [Position of rightmost set bit](https://www.geeksforgeeks.org/position-of-rightmost-set-bit/)
32. [Binary representation of a given number](https://www.geeksforgeeks.org/binary-representation-of-a-given-number/)
33. [Swap all odd and even bits](https://www.geeksforgeeks.org/swap-all-odd-and-even-bits/)
34. [Find position of the only set bit](https://www.geeksforgeeks.org/find-position-of-the-only-set-bit/)
35. [Karatsuba algorithm for fast multiplication](https://www.geeksforgeeks.org/divide-and-conquer-set-2-karatsuba-algorithm-for-fast-multiplication/)
36. [How to swap two numbers without using a temporary variable?](https://www.geeksforgeeks.org/swap-two-numbers-without-using-temporary-variable/)
37. [Check if a number is multiple of 9 using bitwise operators](https://www.geeksforgeeks.org/divisibility-9-using-bitwise-operators/)
38. [Swap two nibbles in a byte](https://www.geeksforgeeks.org/swap-two-nibbles-byte/)
39. [How to turn off a particular bit in a number?](https://www.geeksforgeeks.org/how-to-turn-off-a-particular-bit-in-a-number/)
40. [Check if binary representation of a number is palindrome](https://www.geeksforgeeks.org/check-binary-representation-number-palindrome/)

**Graph Algorithms:**

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

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

1. [Find if the strings can be chained to form a circle](https://www.geeksforgeeks.org/given-array-strings-find-strings-can-chained-form-circle/)
2. [Given a sorted dictionary of an alien language, find order of characters](https://www.geeksforgeeks.org/given-sorted-dictionary-find-precedence-characters/)
3. [Karger’s algorithm for Minimum Cut](https://www.geeksforgeeks.org/kargers-algorithm-for-minimum-cut-set-1-introduction-and-implementation/)
4. [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/)
5. [Hopcroft–Karp Algorithm for Maximum Matching | Set 1 (Introduction)](https://www.geeksforgeeks.org/hopcroft-karp-algorithm-for-maximum-matching-set-1-introduction/)
6. [Hopcroft–Karp Algorithm for Maximum Matching | Set 2 (Implementation)](https://www.geeksforgeeks.org/hopcroft-karp-algorithm-for-maximum-matching-set-1-introduction/)
7. [Length of shortest chain to reach a target word](https://www.geeksforgeeks.org/length-of-shortest-chain-to-reach-a-target-word/)
8. [Find same contacts in a list of contacts](https://www.geeksforgeeks.org/find-same-contacts-in-a-list-of-contacts/)

**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/)

**Quizzes on Algorithms:**

1. [Analysis of Algorithms](http://geeksquiz.com/algorithms/analysis-of-algorithms/)
2. [Sorting](http://geeksquiz.com/algorithms/searching-and-sorting/)
3. [Divide and Conquer](http://geeksquiz.com/algorithms/divide-and-conquer/)
4. [Greedy Algorithms](http://geeksquiz.com/algorithms/greedy-algorithms/)
5. [Dynamic Programming](http://geeksquiz.com/algorithms/dynamic-programming/)
6. [Backtracking](http://geeksquiz.com/algorithms/backtracking/)
7. [Misc](http://geeksquiz.com/algorithms/misc-2/)
8. [NP Complete](http://geeksquiz.com/algorithms/np-complete/)
9. [Searching](http://geeksquiz.com/algorithms/searching/)
10. [Analysis of Algorithms (Recurrences)](http://geeksquiz.com/algorithms/analysis-of-algorithms-recurrences/)
11. [Recursion](http://geeksquiz.com/algorithms/recursion/)
12. [Bit Algorithms](http://geeksquiz.com/algorithms/bit-algorithms/)
13. [Graph Traversals](http://geeksquiz.com/algorithms/graph-traversals/)
14. [Graph Shortest Paths](http://geeksquiz.com/algorithms/graph-shortest-paths/)
15. [Graph Minimum Spanning Tree](http://geeksquiz.com/algorithms/graph-minimum-spanning-tree/)

**Misc:**

1. [Commonly Asked Algorithm Interview Questions | Set 1](http://geeksquiz.com/commonly-asked-algorithm-interview-questions-set-1/)
2. [Given a matrix of ‘O’ and ‘X’, find the largest subsquare surrounded by ‘X’](https://www.geeksforgeeks.org/given-matrix-o-x-find-largest-subsquare-surrounded-x/)
3. [Nuts & Bolts Problem (Lock & Key problem)](https://www.geeksforgeeks.org/nuts-bolts-problem-lock-key-problem/)
4. [Flood fill Algorithm – how to implement fill () in paint?](https://www.geeksforgeeks.org/flood-fill-algorithm-implement-fill-paint/)
5. [Given n appointments, find all conflicting appointments](https://www.geeksforgeeks.org/given-n-appointments-find-conflicting-appointments/)
6. [Check a given sentence for a given set of simple grammer rules](https://www.geeksforgeeks.org/check-given-sentence-given-set-simple-grammer-rules/)
7. [Find Index of 0 to be replaced with 1 to get longest continuous sequence of 1s in a binary array](https://www.geeksforgeeks.org/find-index-0-replaced-1-get-longest-continuous-sequence-1s-binary-array/)
8. [How to check if two given sets are disjoint?](https://www.geeksforgeeks.org/check-two-given-sets-disjoint/)
9. [Minimum Number of Platforms Required for a Railway/Bus Station](https://www.geeksforgeeks.org/minimum-number-platforms-required-railwaybus-station/)
10. [Length of the largest subarray with contiguous elements | Set 1](https://www.geeksforgeeks.org/length-largest-subarray-contiguous-elements-set-1/)
11. [Length of the largest subarray with contiguous elements | Set 2](https://www.geeksforgeeks.org/length-largest-subarray-contiguous-elements-set-2/)
12. [Print all increasing sequences of length k from first n natural numbers](https://www.geeksforgeeks.org/print-increasing-sequences-length-k-first-n-natural-numbers/)
13. [Given two strings, find if first string is a subsequence of second](https://www.geeksforgeeks.org/given-two-strings-find-first-string-subsequence-second/)
14. [Snake and Ladder Problem](https://www.geeksforgeeks.org/snake-ladder-problem-2/)
15. [Write a function that returns 2 for input 1 and returns 1 for 2](http://geeksquiz.com/write-function-returns-2-input-1-returns-1-2/)
16. [Connect n ropes with minimum cost](https://www.geeksforgeeks.org/connect-n-ropes-minimum-cost/)
17. [Find the number of valid parentheses expressions of given length](http://geeksquiz.com/find-number-valid-parentheses-expressions-given-length/)
18. [Longest Monotonically Increasing Subsequence Size (N log N): Simple implementation](http://geeksquiz.com/longest-monotonically-increasing-subsequence-size-n-log-n-simple-implementation/)
19. [Generate all binary permutations such that there are more 1’s than 0’s at every point in all permutations](http://geeksquiz.com/generate-binary-permutations-1s-0s-every-point-permutations/)
20. [Lexicographically minimum string rotation](http://geeksquiz.com/lexicographically-minimum-string-rotation/)
21. [Construct an array from its pair-sum array](http://geeksquiz.com/construct-array-pair-sum-array/)
22. [Program to evaluate simple expressions](http://geeksquiz.com/program-evaluate-simple-expressions/)
23. [Check if characters of a given string can be rearranged to form a palindrome](http://geeksquiz.com/check-characters-given-string-can-rearranged-form-palindrome/)
24. [Print all pairs of anagrams in a given array of strings](http://geeksquiz.com/print-pairs-anagrams-given-array-strings/)