**Array**

[Find pair with given sum in the array](http://www.techiedelight.com/find-pair-with-given-sum-array/)

[Find sub-array with 0 sum](http://www.techiedelight.com/find-sub-array-with-0-sum/)

[Sort binary array in linear time](http://www.techiedelight.com/sort-binary-array-linear-time/)  
[Find a duplicate element in a limited range array](http://www.techiedelight.com/find-duplicate-element-limited-range-array/)  
[Find largest sub-array formed by consecutive integers](http://www.techiedelight.com/find-largest-sub-array-formed-by-consecutive-integers/)  
[Find maximum length sub-array having given sum](http://www.techiedelight.com/find-maximum-length-sub-array-having-given-sum/)  
[Find maximum length sub-array having equal number of 0’s and 1’s](http://www.techiedelight.com/find-maximum-length-sub-array-equal-number-0s-1s/)  
[Sort an array containing 0’s, 1’s and 2’s(Dutch national flag problem)](http://www.techiedelight.com/sort-array-containing-0s-1s-2s-dutch-national-flag-problem/)  
[In place merge two sorted arrays](http://www.techiedelight.com/inplace-merge-two-sorted-arrays/)  
[Merge two arrays by satisfying given constraints](http://www.techiedelight.com/merge-two-arrays-satisfying-given-constraints/)  
[Find index of 0 to replace to get maximum length sequence of continuous ones](http://www.techiedelight.com/find-index-0-replaced-get-maximum-length-sequence-of-continuous-ones/)  
[Find maximum product of two integers in an array](http://www.techiedelight.com/find-maximum-product-two-integers-array/)  
[Shuffle a given array of elements (Fisher–Yates shuffle)](http://www.techiedelight.com/shuffle-given-array-elements-fisher-yates-shuffle/)  
[Rearrange the array with alternate high and low elements](http://www.techiedelight.com/rearrange-the-array-with-alternate-high-and-low-elements/)  
[Find equilibrium index of an array](http://www.techiedelight.com/find-equilibrium-index-array/)  
[Find majority element in an array (Boyer–Moore majority vote algorithm)](http://www.techiedelight.com/find-majority-element-in-an-array-boyer-moore-majority-vote-algorithm/)  
[Move all zeros present in the array to the end](http://www.techiedelight.com/move-zeros-present-array-end/)  
[Replace each element of array with product of every other element without using / operator](http://www.techiedelight.com/replace-element-array-product-every-element-without-using-division-operator/)  
[Find Longest Bitonic Subarray in an array](http://www.techiedelight.com/find-longest-bitonic-subarray-array/)  
[Find maximum difference between two elements in the array by satisfying given constraints](http://www.techiedelight.com/find-maximum-difference-between-two-elements-array/)  
[Maximum subarray problem (Kadane’s algorithm)](http://www.techiedelight.com/maximum-subarray-problem-kadanes-algorithm/)  
[Maximum Sum Circular Subarray](http://www.techiedelight.com/maximum-sum-circular-subarray/)  
[Find all distinct combinations of given length](http://www.techiedelight.com/find-distinct-combinations-of-given-length/)  
[Find all distinct combinations of given length with repetition allowed](http://www.techiedelight.com/find-distinct-combinations-given-length-repetition-allowed/)  
[Find maximum sequence of continuous 1’s formed by replacing at-most k zeroes by ones](http://www.techiedelight.com/find-maximum-sequence-of-continuous-1s-can-formed-replacing-k-zeroes-ones/)  
[Find minimum sum subarray of given size k](http://www.techiedelight.com/find-minimum-sum-subarray-given-size-k/)  
[Find subarray having given sum in given array of integers](http://www.techiedelight.com/find-subarray-having-given-sum-given-array/)  
[Find the length of smallest subarray whose sum of elements is greater than the given number](http://www.techiedelight.com/length-of-smallest-subarray-with-sum-greater-number/)  
[Find largest number possible from set of given numbers](http://www.techiedelight.com/find-largest-number-possible-set-given-numbers/)  
[Find the smallest window in array sorting which will make the entire array sorted](http://www.techiedelight.com/smallest-window-sorting-which-make-array-sorted/)  
[Find maximum sum path involving elements of given arrays](http://www.techiedelight.com/find-maximum-sum-path-involving-elements-given-arrays/)  
[Maximum profit earned by buying and selling shares any number of times](http://www.techiedelight.com/maximum-profit-earned-buying-and-selling-shares)  
[Trapping Rain Water within given set of bars](http://www.techiedelight.com/trapping-rain-water-within-given-set-bars/)  
[Longest Increasing Subsequence](http://www.techiedelight.com/longest-increasing-subsequence/)  
[Find maximum product subarray in a given array](http://www.techiedelight.com/find-maximum-product-subarray-given-array)  
[Find maximum sum of subsequence with no adjacent elements](http://www.techiedelight.com/maximum-sum-of-subsequence-with-no-adjacent-elements)  
[Find minimum platforms needed in the station so to avoid any delay in arrival of any train](http://www.techiedelight.com/minimum-number-of-platforms-needed-avoid-delay-arrival-train)  
[Length of longest continuous sequence with same sum in given binary arrays](http://www.techiedelight.com/length-longest-continuous-sequence-same-sum-binary-arrays/)

[Merging Overlapping Intervals](http://www.techiedelight.com/merging-overlapping-intervals)  
[Activity Selection Problem](http://www.techiedelight.com/activity-selection-problem/)  
[Job Sequencing Problem with Deadlines](http://www.techiedelight.com/job-sequencing-problem-deadlines/)  
[Introduction to Priority Queues using Binary Heaps](http://www.techiedelight.com/introduction-priority-queues-using-binary-heaps/)  
[Min Heap and Max Heap Implementation in C++](http://www.techiedelight.com/min-heap-max-heap-implementation-c/)  
[Heap Sort (Out-of-place and In-place implementation in C++ and C)](http://www.techiedelight.com/heap-sort-place-place-implementation-c-c/)  
[Check if given array represents min heap or not](http://www.techiedelight.com/check-given-array-represents-min-heap-not/)  
[Convert Max Heap to Min Heap in linear time](http://www.techiedelight.com/convert-max-heap-min-heap-linear-time/)  
[Find K’th largest element in an array](http://www.techiedelight.com/find-kth-largest-element-array/)  
[Sort a K-Sorted Array](http://www.techiedelight.com/sort-k-sorted-array/)  
[Merge M sorted lists of variable length](http://www.techiedelight.com/merge-m-sorted-lists-variable-length/)  
[Find K’th smallest element in an array](http://www.techiedelight.com/find-kth-smallest-element-array/)  
[Find smallest range with at-least one element from each of the given lists](http://www.techiedelight.com/find-smallest-range-least-one-element-given-lists/)  
[Merge M sorted lists each containing N elements](http://www.techiedelight.com/merge-m-sorted-lists-containing-n-elements/)  
[Insertion sort | Iterative & Recursive](http://www.techiedelight.com/insertion-sort-iterative-recursive/)  
[Selection sort | Iterative & Recursive](http://www.techiedelight.com/selection-sort-iterative-recursive/)  
[Bubble sort | Iterative & Recursive](http://www.techiedelight.com/bubble-sort-iterative-recursive/)  
[Merge Sort](http://www.techiedelight.com/merge-sort/)  
[Quicksort](http://www.techiedelight.com/quicksort/)  
[Iterative Implementation of Quicksort](http://www.techiedelight.com/iterative-implementation-of-quicksort/)  
[Hybrid QuickSort](http://www.techiedelight.com/hybrid-quicksort/)  
[External merge sort](http://www.techiedelight.com/external-merge-sort/)  
[Custom Sort | Sort elements by their frequency and Index](http://www.techiedelight.com/sort-elements-by-their-frequency-and-index/)  
[Custom Sort | Sort elements of the array by order of elements defined by the second array](http://www.techiedelight.com/custom-sort-sort-elements-array-order-elements-defined-second-array/)  
[Inversion Count of an array](http://www.techiedelight.com/inversion-count-array/)  
[Segregate positive and negative integers in linear time](http://www.techiedelight.com/positive-and-negative-integers-segregate/)  
[Binary Search](http://www.techiedelight.com/binary-search/)  
[Ternary Search vs Binary search](http://www.techiedelight.com/ternary-search-vs-binary-search/)  
[Interpolation search](http://www.techiedelight.com/interpolation-search/)  
[Exponential search](http://www.techiedelight.com/exponential-search/)  
[Find number of rotations in a circularly sorted array](http://www.techiedelight.com/find-number-rotations-circularly-sorted-array/)  
[Search an element in a circular sorted array](http://www.techiedelight.com/search-element-circular-sorted-array/)  
[Find first or last occurrence of a given number in a sorted array](http://www.techiedelight.com/find-first-or-last-occurrence-of-a-given-number-sorted-array/)  
[Count occurrences of a number in a sorted array with duplicates](http://www.techiedelight.com/count-occurrences-number-sorted-array-duplicates/)  
[Find smallest missing element from a sorted array](http://www.techiedelight.com/find-smallest-missing-element-sorted-array/)  
[Find Floor and Ceil of a number in a sorted array](http://www.techiedelight.com/find-floor-ceil-number-sorted-array/)  
[Search in a nearly sorted array in O(logn) time](http://www.techiedelight.com/search-nearly-sorted-array-ologn-time/)  
[Find number of 1’s in a sorted binary array](http://www.techiedelight.com/find-number-1s-sorted-binary-array/)  
[Find the peak element in an array](http://www.techiedelight.com/find-peak-element-array/)  
[Maximum Sum Subarray using Divide & Conquer](http://www.techiedelight.com/maximum-sum-subarray-using-divide-conquer/)  
[Find Minimum and Maximum element in an array using minimum comparisons](http://www.techiedelight.com/find-minimum-maximum-element-array-using-minimum-comparisons/)  
[Matrix Chain Multiplication](http://www.techiedelight.com/matrix-chain-multiplication/)  
[0–1 Knapsack problem](http://www.techiedelight.com/0-1-knapsack-problem/)  
[Maximize value of the expression A[s] — A[r] + A[q] — A[p] where s > r > q > p](http://www.techiedelight.com/maximize-value-of-the-expression/)  
[Partition problem](http://www.techiedelight.com/partition-problem/)  
[Subset sum problem](http://www.techiedelight.com/subset-sum-problem/)  
[Minimum Sum Partition problem](http://www.techiedelight.com/minimum-sum-partition-problem/)  
[Rod Cutting](http://www.techiedelight.com/rot-cutting/)  
[Coin change-making problem (unlimited supply of coins)](http://www.techiedelight.com/coin-change-making-problem-unlimited-supply-coins/)  
[Coin Change Problem — Find total number of ways to get the denomination of coins](http://www.techiedelight.com/coin-change-problem-find-total-number-ways-get-denomination-coins/)  
[Longest alternating subsequence](http://www.techiedelight.com/longest-alternating-subsequence/)  
[Combinations of words formed by replacing given numbers with corresponding English alphabets](http://www.techiedelight.com/combinations-of-words-formed-replacing-given-numbers-corresponding-english-alphabet/)  
[Decode the given sequence to construct minimum number without repeated digits](http://www.techiedelight.com/decode-the-given-sequence-construct-minimum-number-without-repeated-digits/)  
[All combinations of elements satisfiying given constraints](http://www.techiedelight.com/find-combinations-of-elements-satisfies-given-constraints/)

**Backtracking**

[Print all possible solutions to N Queens problem](http://www.techiedelight.com/print-possible-solutions-n-queens-problem/)  
[Print all Possible Knight’s Tours in a chessboard](http://www.techiedelight.com/print-possible-knights-tours-chessboard/)  
[Magnet Puzzle](http://www.techiedelight.com/magnet-puzzle/)  
[Find Shortest Path in Maze](http://www.techiedelight.com/find-shortest-path-in-maze/)  
[Find Longest Possible Route in a Matrix](http://www.techiedelight.com/find-longest-possible-route-matrix/)  
[Find path from source to destination in a matrix that satisfies given constraints](http://www.techiedelight.com/find-path-source-destination-matrix-satisfies-given-constraints/)  
[Find total number of unique paths in a maze from source to destination](http://www.techiedelight.com/find-total-number-unique-paths-maze-source-destination/)  
[Print All Hamiltonian Path present in a graph](http://www.techiedelight.com/print-all-hamiltonian-path-present-in-a-graph/)  
[Print all k-colorable configurations of the graph (Vertex coloring of graph)](http://www.techiedelight.com/print-k-colorable-configurations-graph-vertex-coloring-graph/)  
[Find all Permutations of a given string](http://www.techiedelight.com/find-permutations-given-string/)  
[All combinations of elements satisfiying given constraints](http://www.techiedelight.com/find-combinations-of-elements-satisfies-given-constraints/)  
[Find all binary strings that can be formed from given wildcard pattern](http://www.techiedelight.com/find-binary-strings-can-formed-given-wildcard-pattern/)

**Binary**

[Bit Hacks — Part 1 (Basic)](http://www.techiedelight.com/bit-hacks-part-1-basic/)  
[Bit Hacks — Part 2 (Playing with k’th bit)](http://www.techiedelight.com/bit-hacks-part-2-playing-kth-bit/)  
[Bit Hacks — Part 3 (Playing with rightmost set bit of a number)](http://www.techiedelight.com/bit-hacks-part-3-playing-rightmost-set-bit-number/)  
[Bit Hacks — Part 4 (Playing with letters of English alphabet)](http://www.techiedelight.com/bit-hacks-part-4-playing-letters-english-alphabet/)  
[Bit Hacks — Part 5 (Find absolute value of an integer without branching)](http://www.techiedelight.com/bit-hacks-part-5-find-absolute-value-integer-without-branching/)  
[Bit Hacks — Part 6 (Random Problems)](http://www.techiedelight.com/bit-hacks-part-6-random-problems/)  
[Brian Kernighan’s Algorithm to count set bits in an integer](http://www.techiedelight.com/brian-kernighans-algorithm-count-set-bits-integer/)  
[Compute parity of a number using lookup table](http://www.techiedelight.com/compute-parity-number-using-lookup-table/)  
[Count set bits using lookup table](http://www.techiedelight.com/count-set-bits-using-lookup-table/)  
[Find the minimum or maximum of two integers without using branching](http://www.techiedelight.com/find-minimum-maximum-two-integers-without-using-branching/)  
[Multiply 16-bit integers using 8-bit multiplier](http://www.techiedelight.com/multiply-16-bit-integers-using-8-bit-multiplier/)  
[Round up to the next highest power of 2](http://www.techiedelight.com/round-next-highest-power-2/)  
[Round up to the previous power of 2](http://www.techiedelight.com/round-previous-power-2/)  
[Swap individual bits at given position in an integer](http://www.techiedelight.com/swap-individual-bits-given-position-integer/)  
[Reverse Bits of a given Integer](http://www.techiedelight.com/reverse-bits-of-given-integer/)

[Generate binary numbers between 1 to N](http://www.techiedelight.com/generate-binary-numbers-1-n/)  
[Efficiently implement power function | Recursive and Iterative](http://www.techiedelight.com/power-function-implementation-recursive-iterative/)  
[Find square of a number without using multiplication and division operator | 3 methods](http://www.techiedelight.com/find-square-number-without-using-multiplication-division-operator/)  
[Generate power set of a given set](http://www.techiedelight.com/generate-power-set-given-set/)  
[Huffman Coding](http://www.techiedelight.com/huffman-coding/)

**Binary Tree**

[Check if two given binary trees are identical or not | Iterative & Recursive](http://www.techiedelight.com/check-if-two-binary-trees-are-identical-not-iterative-recursive/)  
[Calculate height of a binary tree | Iterative & Recursive](http://www.techiedelight.com/calculate-height-binary-tree-iterative-recursive/)  
[Delete given Binary Tree | Iterative & Recursive](http://www.techiedelight.com/delete-given-binary-tree-iterative-recursive/)  
[Inorder Tree Traversal | Iterative & Recursive](http://www.techiedelight.com/inorder-tree-traversal-iterative-recursive/)  
[Preorder Tree Traversal | Iterative & Recursive](http://www.techiedelight.com/preorder-tree-traversal-iterative-recursive/)  
[Postorder Tree Traversal | Iterative & Recursive](http://www.techiedelight.com/postorder-tree-traversal-iterative-recursive/)  
[Level Order Traversal of Binary Tree](http://www.techiedelight.com/level-order-traversal-binary-tree/)  
[Spiral Order Traversal of Binary Tree](http://www.techiedelight.com/spiral-order-traversal-binary-tree/)  
[Reverse Level Order Traversal of Binary Tree](http://www.techiedelight.com/reverse-level-order-traversal-binary-tree/)  
[Print all nodes of a given binary tree in specific order](http://www.techiedelight.com/print-nodes-binary-tree-specific-order/)  
[Print left view of binary tree](http://www.techiedelight.com/print-left-view-of-binary-tree/)  
[Print Bottom View of Binary Tree](http://www.techiedelight.com/print-bottom-view-of-binary-tree/)  
[Print Top View of Binary Tree](http://www.techiedelight.com/print-top-view-binary-tree/)  
[Find next node in same level for given node in a binary tree](http://www.techiedelight.com/find-next-node-in-same-level-binary-tree/)  
[Check if given binary tree is complete binary tree or not](http://www.techiedelight.com/check-given-binary-tree-complete-binary-tree-not/)  
[Determine if given two nodes are cousins of each other](http://www.techiedelight.com/determine-two-nodes-are-cousins/)  
[Print cousins of given node in a binary tree](http://www.techiedelight.com/print-cousins-of-given-node-binary-tree/)  
[In-place convert given binary tree to its sum tree](http://www.techiedelight.com/inplace-convert-a-tree-sum-tree/)  
[Check if given binary tree is a sum tree or not](http://www.techiedelight.com/check-given-binary-tree-sum-tree-not/)  
[Combinations of words formed by replacing given numbers with corresponding English alphabets](http://www.techiedelight.com/combinations-of-words-formed-replacing-given-numbers-corresponding-english-alphabet/)  
[Determine if given binary tree is a subtree of another binary tree or not](http://www.techiedelight.com/determine-given-binary-tree-is-subtree-of-another-binary-tree-not/)  
[Find diameter of a binary tree](http://www.techiedelight.com/find-diameter-of-a-binary-tree/)  
[Check if given binary Tree has symmetric structure or not](http://www.techiedelight.com/check-given-binary-tree-symmetric-structure-not/)  
[Convert binary tree to its mirror](http://www.techiedelight.com/convert-binary-tree-to-its-mirror/)  
[Check if binary tree can be converted to another by doing any no. of swaps of left & right child](http://www.techiedelight.com/determine-binary-tree-can-converted-another-number-swaps-left-right-child/)  
[Find Lowest Common Ancestor (LCA) of two nodes in a binary tree](http://www.techiedelight.com/find-lowest-common-ancestor-lca-two-nodes-binary-tree/)  
[Print all paths from root to leaf nodes in given binary tree](http://www.techiedelight.com/print-all-paths-from-root-to-leaf-nodes-binary-tree/)  
[Find ancestors of given node in a Binary Tree](http://www.techiedelight.com/find-ancestors-of-given-node-binary-tree/)  
[Find the distance between given pairs of nodes in a binary tree](http://www.techiedelight.com/distance-between-given-pairs-of-nodes-binary-tree/)  
[Find Vertical Sum in a given Binary Tree](http://www.techiedelight.com/find-vertical-sum-given-binary-tree/)  
[Print nodes in vertical order of a given Binary Tree (Vertical Traversal)](http://www.techiedelight.com/vertical-traversal-binary-tree/)  
[Find the diagonal sum of given binary tree](http://www.techiedelight.com/find-diagonal-sum-given-binary-tree/)  
[Print Diagonal Traversal of Binary Tree](http://www.techiedelight.com/print-diagonal-traversal-binary-tree/)  
[Print corner nodes of every level in binary tree](http://www.techiedelight.com/print-corner-nodes-every-level-binary-tree/)  
[In-place convert convert given Binary Tree to Doubly Linked List](http://www.techiedelight.com/place-convert-given-binary-tree-to-doubly-linked-list/)  
[Sink nodes containing zero to the bottom of the binary tree](http://www.techiedelight.com/sink-nodes-containing-zero-bottom-binary-tree/)  
[Convert given binary tree to full tree by removing half nodes](http://www.techiedelight.com/convert-given-binary-tree-to-full-tree-removing-half-nodes/)  
[Truncate given binary tree to remove nodes which lie on a path having sum less than K](http://www.techiedelight.com/truncate-given-binary-tree-remove-nodes-lie-path-sum-less-k/)  
[Find maximum sum root-to-leaf path in a binary tree](http://www.techiedelight.com/find-maximum-sum-root-to-leaf-path-binary-tree/)  
[Check if given binary tree is height balanced or not](http://www.techiedelight.com/check-given-binary-tree-is-height-balanced-not/)  
[Determine if given Binary Tree is a BST or not](http://www.techiedelight.com/determine-given-binary-tree-is-a-bst-or-not/)

**Binary Search Tree (BST)**

[Insertion in BST](http://www.techiedelight.com/insertion-in-bst/)  
[Search given key in BST](http://www.techiedelight.com/search-given-key-in-bst/)  
[Deletion from BST](http://www.techiedelight.com/deletion-from-bst/)  
[Construct balanced BST from given keys](http://www.techiedelight.com/construct-balanced-bst-given-keys/)  
[Determine if given Binary Tree is a BST or not](http://www.techiedelight.com/determine-given-binary-tree-is-a-bst-or-not/)  
[Check if given keys represents same BSTs or not without building the BST](http://www.techiedelight.com/check-given-keys-represents-same-bsts-not-without-building-bst/)  
[Find inorder predecessor for given key in a BST](http://www.techiedelight.com/find-inorder-predecessor-given-key-bst/)  
[Find Lowest Common Ancestor (LCA) of two nodes in a Binary Search Tree](http://www.techiedelight.com/find-lowest-common-ancestor-lca-two-nodes-bst/)  
[Find K’th smallest and K’th largest element in BST](http://www.techiedelight.com/find-kth-smallest-largest-element-bst/)  
[Floor and Ceil in a Binary Search Tree](http://www.techiedelight.com/floor-ceil-bst-iterative-recursive/)  
[Find optimal cost to construct binary search tree](http://www.techiedelight.com/find-optimal-cost-to-construct-binary-search-tree/)

**Divide & Conquer**

[Binary Search](http://www.techiedelight.com/binary-search/)  
[Ternary Search vs Binary search](http://www.techiedelight.com/ternary-search-vs-binary-search/)  
[Exponential search](http://www.techiedelight.com/exponential-search/)  
[Interpolation search](http://www.techiedelight.com/interpolation-search/)  
[Find number of rotations in a circularly sorted array](http://www.techiedelight.com/find-number-rotations-circularly-sorted-array/)  
[Search an element in a circular sorted array](http://www.techiedelight.com/search-element-circular-sorted-array/)  
[Find first or last occurrence of a given number in a sorted array](http://www.techiedelight.com/find-first-or-last-occurrence-of-a-given-number-sorted-array/)  
[Count occurrences of a number in a sorted array with duplicates](http://www.techiedelight.com/count-occurrences-number-sorted-array-duplicates/)  
[Find smallest missing element from a sorted array](http://www.techiedelight.com/find-smallest-missing-element-sorted-array/)  
[Find Floor and Ceil of a number in a sorted array](http://www.techiedelight.com/find-floor-ceil-number-sorted-array/)  
[Search in a nearly sorted array in O(logn) time](http://www.techiedelight.com/search-nearly-sorted-array-ologn-time/)  
[Find number of 1’s in a sorted binary array](http://www.techiedelight.com/find-number-1s-sorted-binary-array/)  
[Find the peak element in an array](http://www.techiedelight.com/find-peak-element-array/)  
[Maximum Sum Subarray using Divide & Conquer](http://www.techiedelight.com/maximum-sum-subarray-using-divide-conquer/)  
[Find Minimum and Maximum element in an array using minimum comparisons](http://www.techiedelight.com/find-minimum-maximum-element-array-using-minimum-comparisons/)  
[Efficiently implement power function | Recursive and Iterative](http://www.techiedelight.com/power-function-implementation-recursive-iterative/)

[Merge Sort](http://www.techiedelight.com/merge-sort/)  
[Merge Sort for Singly Linked List](http://www.techiedelight.com/merge-sort-singly-linked-list/)  
[Inversion Count of an array](http://www.techiedelight.com/inversion-count-array/)  
[Quicksort](http://www.techiedelight.com/quicksort/)  
[Iterative Implementation of Quicksort](http://www.techiedelight.com/iterative-implementation-of-quicksort/)  
[Hybrid QuickSort](http://www.techiedelight.com/hybrid-quicksort/)

**Dynamic Programming**

[Introduction to Dynamic Programming](http://www.techiedelight.com/introduction-dynamic-programming/)  
[Longest Common Subsequence | Introduction & LCS Length](http://www.techiedelight.com/longest-common-subsequence/)  
[Longest Common Subsequence | Space optimized version](http://www.techiedelight.com/longest-common-subsequence-lcs-space-optimized-version/)  
[Longest Common Subsequence of K-sequences](http://www.techiedelight.com/longest-common-subsequence-of-k-sequences/)  
[Longest Common Subsequence | Finding all LCS](http://www.techiedelight.com/longest-common-subsequence-finding-lcs/)  
[Longest Common Substring problem](http://www.techiedelight.com/longest-common-substring-problem/)  
[Longest Palindromic Subsequence using Dynamic Programming](http://www.techiedelight.com/longest-palindromic-subsequence-using-dynamic-programming/)  
[Longest Repeated Subsequence problem](http://www.techiedelight.com/longest-repeated-subsequence-problem/)  
[Shortest Common Supersequence | Introduction & SCS Length](http://www.techiedelight.com/shortest-common-supersequence-introduction-scs-length/)  
[Shortest Common Supersequence | Finding all SCS](http://www.techiedelight.com/shortest-common-supersequence-finding-scs/)  
[Shortest Common Supersequence | Using LCS](http://www.techiedelight.com/shortest-common-supersequence-using-lcs/)  
[Longest Increasing Subsequence using Dynamic Programming](http://www.techiedelight.com/longest-increasing-subsequence-using-dynamic-programming/)  
[Longest Bitonic Subsequence](http://www.techiedelight.com/longest-bitonic-subsequence/)  
[Increasing Subsequence with Maximum Sum](http://www.techiedelight.com/increasing-subsequence-with-maximum-sum/)  
[The Levenshtein distance (Edit distance) problem](http://www.techiedelight.com/levenshtein-distance-edit-distance-problem/)  
[Find size of largest square sub-matrix of 1’s present in given binary matrix](http://www.techiedelight.com/find-size-largest-square-sub-matrix-1s-present-given-binary-matrix/)  
[Matrix Chain Multiplication](http://www.techiedelight.com/matrix-chain-multiplication/)  
[Find the minimum cost to reach last cell of the matrix from its first cell](http://www.techiedelight.com/find-minimum-cost-reach-last-cell-matrix-first-cell/)  
[Find longest sequence formed by adjacent numbers in the matrix](http://www.techiedelight.com/find-longest-sequence-formed-adjacent-numbers-matrix/)  
[Count number of paths in a matrix with given cost to reach destination cell](http://www.techiedelight.com/counting-paths-on-grid-to-reach-destination-cell/)  
[0–1 Knapsack problem](http://www.techiedelight.com/0-1-knapsack-problem/)  
[Maximize value of the expression A[s] — A[r] + A[q] — A[p] where s > r > q > p](http://www.techiedelight.com/maximize-value-of-the-expression/)  
[Partition problem](http://www.techiedelight.com/partition-problem/)  
[Subset sum problem](http://www.techiedelight.com/subset-sum-problem/)  
[Minimum Sum Partition problem](http://www.techiedelight.com/minimum-sum-partition-problem/)  
[Find all N-digit binary strings without any consecutive 1’s](http://www.techiedelight.com/find-n-digit-binary-strings-without-consecutive-1s/)  
[Rod Cutting](http://www.techiedelight.com/rot-cutting/)  
[Maximum Product Rod Cutting](http://www.techiedelight.com/maximum-product-rod-cutting/)  
[Coin change-making problem (unlimited supply of coins)](http://www.techiedelight.com/coin-change-making-problem-unlimited-supply-coins/)  
[Coin Change Problem — Find total number of ways to get the denomination of coins](http://www.techiedelight.com/coin-change-problem-find-total-number-ways-get-denomination-coins/)  
[Longest alternating subsequence](http://www.techiedelight.com/longest-alternating-subsequence/)  
[Count number of times a pattern appears in given string as a subsequence](http://www.techiedelight.com/count-number-times-pattern-appears-given-string-subsequence/)  
[Collect maximum points in a matrix by satisfying given constraints](http://www.techiedelight.com/collect-maximum-points-matrix-satisfying-given-constraints/)  
[Count total possible combinations of N-digit numbers in a mobile keypad](http://www.techiedelight.com/count-total-possible-combinations-n-digit-numbers-mobile-keypad/)  
[Find optimal cost to construct binary search tree](http://www.techiedelight.com/find-optimal-cost-to-construct-binary-search-tree/)  
[Word Break Problem](http://www.techiedelight.com/word-break-problem/)  
[Wildcard Pattern Matching](http://www.techiedelight.com/wildcard-pattern-matching/)

[Find probability that a person is alive after taking N steps on the island](http://www.techiedelight.com/probability-alive-after-taking-n-steps-island/)  
[Calculate sum of all elements in a sub-matrix in constant time](http://www.techiedelight.com/calculate-sum-elements-sub-matrix-constant-time/)  
[Find maximum sum K x K sub-matrix in a given M x N matrix](http://www.techiedelight.com/find-maximum-sum-submatrix-in-given-matrix/)  
[Find maximum sum submatrix present in a given matrix](http://www.techiedelight.com/find-maximum-sum-submatrix-present-given-matrix/)  
[Find maximum sum of subsequence with no adjacent elements](http://www.techiedelight.com/maximum-sum-of-subsequence-with-no-adjacent-elements)  
[Maximum subarray problem (Kadane’s algorithm)](http://www.techiedelight.com/maximum-subarray-problem-kadanes-algorithm/)  
[Single-Source Shortest Paths — Bellman Ford Algorithm](http://www.techiedelight.com/single-source-shortest-paths-bellman-ford-algorithm/)  
[All-Pairs Shortest Paths — Floyd Warshall Algorithm](http://www.techiedelight.com/pairs-shortest-paths-floyd-warshall-algorithm/)

**Graphs**

[Terminology and Representations of Graphs](http://www.techiedelight.com/terminology-and-representations-of-graphs/)  
[Graph Implementation using STL](http://www.techiedelight.com/graph-implementation-using-stl/)  
[Graph Implementation in C++ without using STL](http://www.techiedelight.com/graph-implementation-c-without-using-stl/)  
[Breadth First Search (BFS) | Iterative & Recursive Implementation](http://www.techiedelight.com/breadth-first-search/)  
[Depth First Search (DFS) | Iterative & Recursive Implementation](http://www.techiedelight.com/depth-first-search/)  
[Arrival and Departure Time of Vertices in DFS](http://www.techiedelight.com/arrival-departure-time-vertices-dfs/)  
[Types of edges involved in DFS and relation between them](http://www.techiedelight.com/types-edges-involved-dfs-relation/)  
[Bipartite Graph](http://www.techiedelight.com/bipartite-graph/)  
[Minimum number of throws required to win Snake and Ladder game](http://www.techiedelight.com/min-throws-required-to-win-snake-and-ladder-game/)  
[Topological Sorting in a DAG](http://www.techiedelight.com/topological-sorting-dag/)  
[Transitive Closure of a Graph](http://www.techiedelight.com/transitive-closure-graph/)  
[Check if an undirected graph contains cycle or not](http://www.techiedelight.com/check-undirected-graph-contains-cycle-not/)  
[Total number of paths in given digraph from given source to destination having exactly m edges](http://www.techiedelight.com/total-paths-in-digraph-from-source-to-destination-m-edges/)  
[Determine if an undirected graph is a Tree (Acyclic Connected Graph)](http://www.techiedelight.com/determine-undirected-graph-tree-acyclic-connected-graph/)  
[2-Edge Connectivity in the graph](http://www.techiedelight.com/2-edge-connectivity-graph/)  
[2-Vertex Connectivity in the graph](http://www.techiedelight.com/2-vertex-connectivity-graph/)  
[Check if given digraph is a DAG (Directed Acyclic Graph) or not](http://www.techiedelight.com/check-given-digraph-dag-directed-acyclic-graph-not/)  
[Disjoint-Set Data Structure (Union-Find Algorithm)](http://www.techiedelight.com/disjoint-set-data-structure-union-find-algorithm/)  
[Chess Knight Problem — Find Shortest path from source to destination](http://www.techiedelight.com/chess-knight-problem-find-shortest-path-source-destination/)  
[Check if given Graph is Strongly Connected or not](http://www.techiedelight.com/check-given-graph-strongly-connected-not/)  
[Check if given Graph is Strongly Connected or not using one DFS Traversal](http://www.techiedelight.com/check-graph-strongly-connected-one-dfs-traversal/)  
[Union-Find Algorithm for Cycle Detection in undirected graph](http://www.techiedelight.com/union-find-algorithm-cycle-detection-graph/)  
[Kruskal’s Algorithm for finding Minimum Spanning Tree](http://www.techiedelight.com/kruskals-algorithm-for-finding-minimum-spanning-tree/)  
[Single-Source Shortest Paths — Dijkstra’s Algorithm](http://www.techiedelight.com/single-source-shortest-paths-dijkstras-algorithm/)  
[Single-Source Shortest Paths — Bellman Ford Algorithm](http://www.techiedelight.com/single-source-shortest-paths-bellman-ford-algorithm/)  
[All-Pairs Shortest Paths — Floyd Warshall Algorithm](http://www.techiedelight.com/pairs-shortest-paths-floyd-warshall-algorithm/)

[Print all k-colorable configurations of the graph (Vertex coloring of graph)](http://www.techiedelight.com/print-k-colorable-configurations-graph-vertex-coloring-graph/)  
[Print All Hamiltonian Path present in a graph](http://www.techiedelight.com/print-all-hamiltonian-path-present-in-a-graph/)  
[Greedy coloring of graph](http://www.techiedelight.com/greedy-coloring-graph/)

**Heaps**

[Introduction to Priority Queues using Binary Heaps](http://www.techiedelight.com/introduction-priority-queues-using-binary-heaps/)  
[Min Heap and Max Heap Implementation in C++](http://www.techiedelight.com/min-heap-max-heap-implementation-c/)  
[Heap Sort (Out-of-place and In-place implementation in C++ and C)](http://www.techiedelight.com/heap-sort-place-place-implementation-c-c/)  
[Check if given array represents min heap or not](http://www.techiedelight.com/check-given-array-represents-min-heap-not/)  
[Convert Max Heap to Min Heap in linear time](http://www.techiedelight.com/convert-max-heap-min-heap-linear-time/)  
[Find K’th largest element in an array](http://www.techiedelight.com/find-kth-largest-element-array/)  
[Sort a K-Sorted Array](http://www.techiedelight.com/sort-k-sorted-array/)  
[Merge M sorted lists of variable length](http://www.techiedelight.com/merge-m-sorted-lists-variable-length/)  
[Find K’th smallest element in an array](http://www.techiedelight.com/find-kth-smallest-element-array/)  
[Find smallest range with at-least one element from each of the given lists](http://www.techiedelight.com/find-smallest-range-least-one-element-given-lists/)  
[Merge M sorted lists each containing N elements](http://www.techiedelight.com/merge-m-sorted-lists-containing-n-elements/)

[External merge sort](http://www.techiedelight.com/external-merge-sort/)  
[Huffman Coding](http://www.techiedelight.com/huffman-coding/)  
[Find first k maximum occurring words in given set of strings](http://www.techiedelight.com/find-first-k-maximum-occurring-words-given-set-strings/)  
[Find first k non-repeating characters in a string in single traversal](http://www.techiedelight.com/first-k-non-repeating-characters-string/)

**Linked List**

[Introduction to Linked Lists](http://www.techiedelight.com/introduction-linked-lists/)  
[Linked List Implementation | Part 1](http://www.techiedelight.com/linked-list-implementation-part-1/)  
[Linked List Implementation | Part 2](http://www.techiedelight.com/linked-list-implementation-part-2/)  
[Static Linked List in C](http://www.techiedelight.com/static-linked-list-c/)  
[Clone given Linked List](http://www.techiedelight.com/clone-given-linked-list/)  
[Delete Linked List](http://www.techiedelight.com/delete-linked-list/)  
[Pop operation in linked list](http://www.techiedelight.com/pop-operation-in-linked-list/)  
[Insert given node into the correct sorted position in the given sorted linked list](http://www.techiedelight.com/sorted-insert-in-linked-list/)  
[Given a linked list, change it to be in sorted order](http://www.techiedelight.com/given-linked-list-change-sorted-order/)  
[Split the nodes of the given linked list into front and back halves](http://www.techiedelight.com/split-nodes-given-linked-list-front-back-halves/)  
[Remove duplicates from a sorted linked list](http://www.techiedelight.com/remove-duplicates-sorted-linked-list/)  
[Move front node of the given list to the front of the another list](http://www.techiedelight.com/move-front-node-given-list-front-another-list/)  
[Move even nodes to the end of the list in reverse order](http://www.techiedelight.com/move-even-nodes-to-end-of-list-in-reverse-order/)  
[Split given linked list into two lists where each list containing alternating elements from it](http://www.techiedelight.com/split-linked-list-into-two-lists-list-containing-alternating-elements/)  
[Construct a linked list by merging alternate nodes of two given lists](http://www.techiedelight.com/merge-alternate-nodes-two-linked-lists/)  
[Merge given sorted linked lists into one](http://www.techiedelight.com/merge-given-sorted-linked-lists/)  
[Merge Sort for Singly Linked List](http://www.techiedelight.com/merge-sort-singly-linked-list/)  
[Intersection of two given sorted linked lists](http://www.techiedelight.com/intersection-two-given-sorted-linked-lists/)  
[Reverse linked list | Part 1 (Iterative Solution)](http://www.techiedelight.com/reverse-linked-list-part-1-iterative-solution/)  
[Reverse linked list | Part 2 (Recursive Solution)](http://www.techiedelight.com/reverse-linked-list-part-2-recursive-solution/)  
[Reverse every group of k nodes in given linked list](http://www.techiedelight.com/reverse-every-k-nodes-of-a-linked-list/)  
[Find K’th node from the end in a linked list](http://www.techiedelight.com/find-kth-node-from-the-end-linked-list/)  
[Merge alternate nodes of two linked lists into the first list](http://www.techiedelight.com/merge-alternate-nodes-two-linked-lists-first-list/)  
[Merge two sorted linked lists from their end](http://www.techiedelight.com/merge-two-sorted-linked-lists-end/)  
[Delete every N nodes in a linked list after skipping M nodes](http://www.techiedelight.com/delete-every-n-nodes-linked-list-skipping-m-nodes/)  
[Rearrange linked list in specific manner in linear time](http://www.techiedelight.com/rearrange-linked-list-specific-manner-linear-time/)  
[Check if linked list is palindrome or not](http://www.techiedelight.com/check-if-linked-list-is-palindrome/)  
[Move last node to front in a given Linked List](http://www.techiedelight.com/move-last-node-to-front-linked-list/)  
[Rearrange the linked list in specific manner](http://www.techiedelight.com/rearrange-the-linked-list-specific-manner/)  
[Detect Cycle in a linked list (Floyd’s Cycle Detection Algorithm)](http://www.techiedelight.com/detect-cycle-linked-list-floyds-cycle-detection-algorithm/)

**Matrix**

[Print Matrix in Spiral Order](http://www.techiedelight.com/print-matrix-spiral-order/)  
[Create Spiral Matrix from given array](http://www.techiedelight.com/create-spiral-matrix-given-array/)  
[Shift all matrix elements by 1 in Spiral Order](http://www.techiedelight.com/shift-matrix-elements-1-spiral-order/)  
[Find Shortest path from source to destination in a matrix that satisfies given constraints](http://www.techiedelight.com/find-shortest-path-source-destination-matrix-satisfies-given-constraints/)  
[Change all elements of row i and column j in a matrix to 0 if cell (i, j) has value 0](http://www.techiedelight.com/change-elements-row-column-j-matrix-0-cell-j-value-0/)  
[Print diagonal elements of the matrix having positive slope](http://www.techiedelight.com/print-matrix-diagonally-positive-slope/)  
[Find all paths from first cell to last cell of a matrix](http://www.techiedelight.com/find-all-paths-from-source-to-destination-in-matrix/)  
[Replace all occurrences of 0 that are not surrounded by 1 in a binary matrix](http://www.techiedelight.com/replace-occurrences-0-not-surrounded-1-binary-matrix/)  
[In-place rotate the matrix by 90 degrees in clock-wise direction](http://www.techiedelight.com/place-rotate-matrix-90-degrees-clock-wise-direction/)  
[Count negative elements present in sorted matrix in linear time](http://www.techiedelight.com/count-negative-elements-present-sorted-matrix/)  
[Report all occurrences of an element in row wise and column wise sorted matrix in linear time](http://www.techiedelight.com/report-all-occurrences-of-an-element-in-sorted-matrix/)  
[Calculate sum of all elements in a sub-matrix in constant time](http://www.techiedelight.com/calculate-sum-elements-sub-matrix-constant-time/)  
[Find maximum sum K x K sub-matrix in a given M x N matrix](http://www.techiedelight.com/find-maximum-sum-submatrix-in-given-matrix/)  
[Find maximum sum submatrix present in a given matrix](http://www.techiedelight.com/find-maximum-sum-submatrix-present-given-matrix/)  
[Find probability that a person is alive after taking N steps on the island](http://www.techiedelight.com/probability-alive-after-taking-n-steps-island/)  
[Count the number of islands](http://www.techiedelight.com/count-the-number-of-islands/)  
[Flood fill Algorithm](http://www.techiedelight.com/flood-fill-algorithm/)  
[Find shortest safe route in a field with sensors present](http://www.techiedelight.com/find-shortest-safe-route-field-sensors-present/)  
[Find all occurrences of given string in a character matrix](http://www.techiedelight.com/find-occurrences-given-string-character-matrix/)  
[Lee algorithm | Shortest path in a Maze](http://www.techiedelight.com/lee-algorithm-shortest-path-in-a-maze/)

[Travelling Salesman Problem using Branch and Bound](http://www.techiedelight.com/travelling-salesman-problem-using-branch-and-bound/)  
[Collect maximum points in a matrix by satisfying given constraints](http://www.techiedelight.com/collect-maximum-points-matrix-satisfying-given-constraints/)  
[Count number of paths in a matrix with given cost to reach destination cell](http://www.techiedelight.com/counting-paths-on-grid-to-reach-destination-cell/)  
[Find longest sequence formed by adjacent numbers in the matrix](http://www.techiedelight.com/find-longest-sequence-formed-adjacent-numbers-matrix/)  
[Find the minimum cost to reach last cell of the matrix from its first cell](http://www.techiedelight.com/find-minimum-cost-reach-last-cell-matrix-first-cell/)  
[Matrix Chain Multiplication](http://www.techiedelight.com/matrix-chain-multiplication/)  
[Find size of largest square sub-matrix of 1’s present in given binary matrix](http://www.techiedelight.com/find-size-largest-square-sub-matrix-1s-present-given-binary-matrix/)  
[Chess Knight Problem — Find Shortest path from source to destination](http://www.techiedelight.com/chess-knight-problem-find-shortest-path-source-destination/)  
[Find Duplicate rows in a binary matrix](http://www.techiedelight.com/find-duplicate-rows-binary-matrix/)  
[Print all possible solutions to N Queens problem](http://www.techiedelight.com/print-possible-solutions-n-queens-problem/)  
[Print all Possible Knight’s Tours in a chessboard](http://www.techiedelight.com/print-possible-knights-tours-chessboard/)  
[Find Shortest Path in Maze](http://www.techiedelight.com/find-shortest-path-in-maze/)  
[Find Longest Possible Route in a Matrix](http://www.techiedelight.com/find-longest-possible-route-matrix/)

**Queue**

[Chess Knight Problem — Find Shortest path from source to destination](http://www.techiedelight.com/chess-knight-problem-find-shortest-path-source-destination/)  
[Lee algorithm | Shortest path in a Maze](http://www.techiedelight.com/lee-algorithm-shortest-path-in-a-maze/)  
[Find shortest safe route in a field with sensors present](http://www.techiedelight.com/find-shortest-safe-route-field-sensors-present/)  
[Flood fill Algorithm](http://www.techiedelight.com/flood-fill-algorithm/)  
[Count the number of islands](http://www.techiedelight.com/count-the-number-of-islands/)  
[Find Shortest path from source to destination in a matrix that satisfies given constraints](http://www.techiedelight.com/find-shortest-path-source-destination-matrix-satisfies-given-constraints/)  
[Generate binary numbers between 1 to N](http://www.techiedelight.com/generate-binary-numbers-1-n/)  
[Calculate height of a binary tree | Iterative & Recursive](http://www.techiedelight.com/calculate-height-binary-tree-iterative-recursive/)  
[Delete given Binary Tree | Iterative & Recursive](http://www.techiedelight.com/delete-given-binary-tree-iterative-recursive/)  
[Level Order Traversal of Binary Tree](http://www.techiedelight.com/level-order-traversal-binary-tree/)  
[Spiral Order Traversal of Binary Tree](http://www.techiedelight.com/spiral-order-traversal-binary-tree/)  
[Reverse Level Order Traversal of Binary Tree](http://www.techiedelight.com/reverse-level-order-traversal-binary-tree/)  
[Print all nodes of a given binary tree in specific order](http://www.techiedelight.com/print-nodes-binary-tree-specific-order/)  
[Print left view of binary tree](http://www.techiedelight.com/print-left-view-of-binary-tree/)  
[Find next node in same level for given node in a binary tree](http://www.techiedelight.com/find-next-node-in-same-level-binary-tree/)  
[Check if given binary tree is complete binary tree or not](http://www.techiedelight.com/check-given-binary-tree-complete-binary-tree-not/)  
[Print Diagonal Traversal of Binary Tree](http://www.techiedelight.com/print-diagonal-traversal-binary-tree/)  
[Print corner nodes of every level in binary tree](http://www.techiedelight.com/print-corner-nodes-every-level-binary-tree/)  
[Breadth First Search (BFS) | Iterative & Recursive Implementation](http://www.techiedelight.com/breadth-first-search/)  
[Minimum number of throws required to win Snake and Ladder game](http://www.techiedelight.com/min-throws-required-to-win-snake-and-ladder-game/)  
[Check if an undirected graph contains cycle or not](http://www.techiedelight.com/check-undirected-graph-contains-cycle-not/)

**Sorting**

[Insertion sort | Iterative & Recursive](http://www.techiedelight.com/insertion-sort-iterative-recursive/)  
[Selection sort | Iterative & Recursive](http://www.techiedelight.com/selection-sort-iterative-recursive/)  
[Bubble sort | Iterative & Recursive](http://www.techiedelight.com/bubble-sort-iterative-recursive/)  
[Merge Sort](http://www.techiedelight.com/merge-sort/)  
[Quicksort](http://www.techiedelight.com/quicksort/)  
[Iterative Implementation of Quicksort](http://www.techiedelight.com/iterative-implementation-of-quicksort/)  
[Hybrid QuickSort](http://www.techiedelight.com/hybrid-quicksort/)  
[External merge sort](http://www.techiedelight.com/external-merge-sort/)  
[Custom Sort | Sort elements by their frequency and Index](http://www.techiedelight.com/sort-elements-by-their-frequency-and-index/)  
[Custom Sort | Sort elements of the array by order of elements defined by the second array](http://www.techiedelight.com/custom-sort-sort-elements-array-order-elements-defined-second-array/)  
[Inversion Count of an array](http://www.techiedelight.com/inversion-count-array/)  
[Segregate positive and negative integers in linear time](http://www.techiedelight.com/positive-and-negative-integers-segregate/)

[Find the smallest window in array sorting which will make the entire array sorted](http://www.techiedelight.com/smallest-window-sorting-which-make-array-sorted/)  
[Find largest number possible from set of given numbers](http://www.techiedelight.com/find-largest-number-possible-set-given-numbers/)  
[Move all zeros present in the array to the end](http://www.techiedelight.com/move-zeros-present-array-end/)  
[Sort binary array in linear time](http://www.techiedelight.com/sort-binary-array-linear-time/)  
[Merge Sort for Singly Linked List](http://www.techiedelight.com/merge-sort-singly-linked-list/)  
[Group anagrams together from given list of words](http://www.techiedelight.com/group-anagrams-together-given-list-words/)  
[Activity Selection Problem](http://www.techiedelight.com/activity-selection-problem/)  
[Lexicographic sorting of given set of keys](http://www.techiedelight.com/lexicographic-sorting-given-set-of-keys/)  
[Heap Sort (Out-of-place and In-place implementation in C++ and C)](http://www.techiedelight.com/heap-sort-place-place-implementation-c-c/)  
[Merge M sorted lists of variable length](http://www.techiedelight.com/merge-m-sorted-lists-variable-length/)  
[Merge M sorted lists each containing N elements](http://www.techiedelight.com/merge-m-sorted-lists-containing-n-elements/)  
[Find all palindromic permutations of a string](http://www.techiedelight.com/find-palindromic-permutations-string/)  
[Find all lexicographically next permutations of a string sorted in ascending order](http://www.techiedelight.com/find-lexicographically-next-permutations-string-sorted-ascending-order/)  
[Merge two sorted linked lists from their end](http://www.techiedelight.com/merge-two-sorted-linked-lists-end/)  
[Sort an array containing 0’s, 1’s and 2’s (Dutch national flag problem)](http://www.techiedelight.com/sort-array-containing-0s-1s-2s-dutch-national-flag-problem/)  
[Find pair with given sum in the array](http://www.techiedelight.com/find-pair-with-given-sum-array/)  
[Inplace merge two sorted arrays](http://www.techiedelight.com/inplace-merge-two-sorted-arrays/)  
[Merge two arrays by satisfying given constraints](http://www.techiedelight.com/merge-two-arrays-satisfying-given-constraints/)  
[Find maximum product of two integers in an array](http://www.techiedelight.com/find-maximum-product-two-integers-array/)  
[Find all distinct combinations of given length](http://www.techiedelight.com/find-distinct-combinations-of-given-length/)  
[Find all distinct combinations of given length with repetition allowed](http://www.techiedelight.com/find-distinct-combinations-given-length-repetition-allowed/)  
[Merging Overlapping Intervals](http://www.techiedelight.com/merging-overlapping-intervals)

**Stack**

[Check if given expression is balanced expression or not](http://www.techiedelight.com/check-given-expression-balanced-expression-not/)  
[Find duplicate parenthesis in an expression](http://www.techiedelight.com/find-duplicate-parenthesis-expression/)  
[Evaluate given postfix expression](http://www.techiedelight.com/evaluate-given-postfix-expression/)  
[Decode the given sequence to construct minimum number without repeated digits](http://www.techiedelight.com/decode-the-given-sequence-construct-minimum-number-without-repeated-digits/)

[Inorder Tree Traversal | Iterative & Recursive](http://www.techiedelight.com/inorder-tree-traversal-iterative-recursive/)  
[Preorder Tree Traversal | Iterative & Recursive](http://www.techiedelight.com/preorder-tree-traversal-iterative-recursive/)  
[Postorder Tree Traversal | Iterative & Recursive](http://www.techiedelight.com/postorder-tree-traversal-iterative-recursive/)  
[Find ancestors of given node in a Binary Tree](http://www.techiedelight.com/find-ancestors-of-given-node-binary-tree/)  
[Check if two given binary trees are identical or not | Iterative & Recursive](http://www.techiedelight.com/check-if-two-binary-trees-are-identical-not-iterative-recursive/)  
[Reverse given text without reversing the individual words](http://www.techiedelight.com/reverse-text-without-reversing-individual-words/)  
[Find all binary strings that can be formed from given wildcard pattern](http://www.techiedelight.com/find-binary-strings-can-formed-given-wildcard-pattern/)  
[Iterative Implementation of Quicksort](http://www.techiedelight.com/iterative-implementation-of-quicksort/)  
[Depth First Search (DFS) | Iterative & Recursive Implementation](http://www.techiedelight.com/depth-first-search/)

**String**

[Check if given set of moves is circular or not](http://www.techiedelight.com/check-given-set-moves-circular-not/)  
[Check if given string is a rotated palindrome or not](http://www.techiedelight.com/check-given-string-rotated-palindrome-not/)  
[Longest Palindromic Substring (Non-DP Space Optimized Solution)](http://www.techiedelight.com/longest-palindromic-substring-non-dp-space-optimized-solution/)  
[Check if repeated subsequence is present in the string or not](http://www.techiedelight.com/check-repeated-subsequence-present-string-not/)  
[Check if strings can be derived from each other by circularly rotating them](http://www.techiedelight.com/check-strings-can-derived-circularly-rotating/)  
[Convert given number into corresponding excel column name](http://www.techiedelight.com/convert-given-number-corresponding-excel-column-name/)  
[Determine if two strings are anagram or not](http://www.techiedelight.com/determine-if-two-strings-are-anagram-or-not/)  
[Find all binary strings that can be formed from given wildcard pattern](http://www.techiedelight.com/find-binary-strings-can-formed-given-wildcard-pattern/)  
[Find all interleavings of given strings](http://www.techiedelight.com/find-interleavings-of-given-strings/)  
[Isomorphic Strings](http://www.techiedelight.com/isomorphic-strings/)  
[Find all possible palindromic substrings in a string](http://www.techiedelight.com/find-possible-palindromic-substrings-string/)  
[Find all possible combinations of words formed from mobile keypad](http://www.techiedelight.com/find-possible-combinations-words-formed-from-mobile-keypad/)  
[Find all possible combinations by replacing given digits with characters of the corresponding list](http://www.techiedelight.com/possible-combinations-replacing-given-digits-corresponding-list/)  
[Find all words from given list that follows same order of characters as given pattern](http://www.techiedelight.com/find-words-that-follows-given-pattern/)  
[Find first k non-repeating characters in a string in single traversal](http://www.techiedelight.com/first-k-non-repeating-characters-string/)  
[Group anagrams together from given list of words](http://www.techiedelight.com/group-anagrams-together-given-list-words/)  
[Introduction to Pattern Matching](http://www.techiedelight.com/introduction-pattern-matching/)  
[Inplace remove all occurrences of ‘AB’ and ‘C’ from the string](http://www.techiedelight.com/inplace-remove-all-occurrences-ab-c-string/)  
[Longest even length palidromic sum substring](http://www.techiedelight.com/longest-even-length-palidromic-sum-substring/)  
[Print string in zig-zag form in k rows](http://www.techiedelight.com/print-string-in-zig-zag-form-k-rows/)  
[Reverse given text without reversing the individual words](http://www.techiedelight.com/reverse-text-without-reversing-individual-words/)  
[Run Length Encoding (RLE) data compression algorithm](http://www.techiedelight.com/run-length-encoding-rle-data-compression-algorithm/)  
[Validate an IP address](http://www.techiedelight.com/validate-ip-address/)  
[Find the longest substring of given string containing k distinct characters](http://www.techiedelight.com/find-longest-substring-containing-k-distinct-characters/)  
[Find all palindromic permutations of a string](http://www.techiedelight.com/find-palindromic-permutations-string/)  
[Find all substrings of a string that are permutation of a given string](http://www.techiedelight.com/find-substrings-string-permutation-given-string/)  
[Find the longest substring of given string containing all distinct characters](http://www.techiedelight.com/find-longest-substring-given-string-containing-distinct-characters/)  
[Find all Permutations of a given string](http://www.techiedelight.com/find-permutations-given-string/)  
[Find all lexicographically next permutations of a string sorted in ascending order](http://www.techiedelight.com/find-lexicographically-next-permutations-string-sorted-ascending-order/)  
[Find Lexicographically minimal string rotation](http://www.techiedelight.com/find-lexicographically-minimal-string-rotation/)  
[Find all strings of given length containing balanced parentheses](http://www.techiedelight.com/find-strings-given-length-containing-balanced-parentheses/)  
[Find all N-digit binary numbers with k-bits set where k ranges from 1 to N](http://www.techiedelight.com/find-n-digit-binary-numbers-k-bits-set-k-ranges-1-n/)  
[Generate binary numbers between 1 to N](http://www.techiedelight.com/generate-binary-numbers-1-n/)  
[Find all combinations of non-overlapping substrings of a string](http://www.techiedelight.com/find-combinations-non-overlapping-substrings-string/)  
[Check if given sentence is syntactically correct or not](http://www.techiedelight.com/check-given-sentence-syntactically-correct-not/)  
[Find all N-digit strictly increasing numbers (Bottom-Up and Top-Down Approach)](http://www.techiedelight.com/find-n-digit-strictly-increasing-numbers-bottom-top-approach/)

[Combinations of words formed by replacing given numbers with corresponding English alphabets](http://www.techiedelight.com/combinations-of-words-formed-replacing-given-numbers-corresponding-english-alphabet/)  
[Word Break Problem](http://www.techiedelight.com/word-break-problem/)  
[Wildcard Pattern Matching](http://www.techiedelight.com/wildcard-pattern-matching/)  
[Count number of times a pattern appears in given string as a subsequence](http://www.techiedelight.com/count-number-times-pattern-appears-given-string-subsequence/)  
[The Levenshtein distance (Edit distance) problem](http://www.techiedelight.com/levenshtein-distance-edit-distance-problem/)  
[Longest Common Subsequence | Introduction & LCS Length](http://www.techiedelight.com/longest-common-subsequence/)  
[Longest Common Subsequence | Space optimized version](http://www.techiedelight.com/longest-common-subsequence-lcs-space-optimized-version/)  
[Longest Common Subsequence of K-sequences](http://www.techiedelight.com/longest-common-subsequence-of-k-sequences/)  
[Longest Common Subsequence | Finding all LCS](http://www.techiedelight.com/longest-common-subsequence-finding-lcs/)  
[Longest Repeated Subsequence problem](http://www.techiedelight.com/longest-repeated-subsequence-problem/)  
[Longest Palindromic Subsequence using Dynamic Programming](http://www.techiedelight.com/longest-palindromic-subsequence-using-dynamic-programming/)  
[Longest Common Substring problem](http://www.techiedelight.com/longest-common-substring-problem/)  
[Shortest Common Supersequence | Introduction & SCS Length](http://www.techiedelight.com/shortest-common-supersequence-introduction-scs-length/)  
[Shortest Common Supersequence | Finding all SCS](http://www.techiedelight.com/shortest-common-supersequence-finding-scs/)  
[Shortest Common Supersequence | Using LCS](http://www.techiedelight.com/shortest-common-supersequence-using-lcs/)

**Trie**

[Trie Implementation | Insert, Search and Delete](http://www.techiedelight.com/trie-implementation-insert-search-delete/)  
[Memory efficient Trie Implementation using Map | Insert, Search and Delete](http://www.techiedelight.com/memory-efficient-trie-implementation-using-map-insert-search-delete/)  
[Longest Common Prefix in given set of strings (using Trie)](http://www.techiedelight.com/longest-common-prefix-given-set-strings-using-trie/)  
[Lexicographic sorting of given set of keys](http://www.techiedelight.com/lexicographic-sorting-given-set-of-keys/)  
[Find maximum occurring word in given set of strings](http://www.techiedelight.com/find-maximum-occurring-word-given-set-strings/)  
[Find first k maximum occurring words in given set of strings](http://www.techiedelight.com/find-first-k-maximum-occurring-words-given-set-strings/)  
[Find Duplicate rows in a binary matrix](http://www.techiedelight.com/find-duplicate-rows-binary-matrix/)

**Greedy**

[Activity Selection Problem](http://www.techiedelight.com/activity-selection-problem/)  
[Huffman Coding](http://www.techiedelight.com/huffman-coding/)  
[Shortest Superstring Problem](http://www.techiedelight.com/shortest-superstring-problem/)  
[Job Sequencing Problem with Deadlines](http://www.techiedelight.com/job-sequencing-problem-deadlines/)  
[Greedy coloring of graph](http://www.techiedelight.com/greedy-coloring-graph/)

**Puzzles**

[Clock angle problem — Find angle between hour and minute hand](http://www.techiedelight.com/angle-between-hour-minute-hand/)  
[Add two numbers without using addition operator | 5 methods](http://www.techiedelight.com/add-two-numbers-without-using-addition-operator/)  
[Generate power set of a given set](http://www.techiedelight.com/generate-power-set-given-set/)  
[Implement power function without using multiplication and division operators](http://www.techiedelight.com/implement-power-function-without-using-multiplication-division-operators/)  
[Print all numbers between 1 to N without using semicolon](http://www.techiedelight.com/print-numbers-1-n-without-using-semicolon/)  
[Swap two numbers without using third variable | 5 methods](http://www.techiedelight.com/swap-two-numbers-without-using-third-variable/)  
[Determine the if condition to print specific output](http://www.techiedelight.com/determine-condition-to-print-specific-output/)  
[Find maximum, minimum of three numbers without using conditional statement and ternary operator | 4 methods](http://www.techiedelight.com/maximum-minimum-three-numbers-without-using-conditional-statement-ternary-operator/)  
[Find numbers represented as sum of two cubes for two different pairs](http://www.techiedelight.com/numbers-represented-as-sum-of-two-cubes/)  
[Print “Hello World” with empty main() function | 3 methods](http://www.techiedelight.com/print-hello-world-empty-main-function/)  
[Tower of Hanoi Problem](http://www.techiedelight.com/tower-of-hanoi-problem/)  
[Print all numbers between 1 to N without using any loop | 4 methods](http://www.techiedelight.com/print-numbers-1-n-without-using-loop-4-methods/)  
[Print a semicolon without using semicolon anywhere in the program](http://www.techiedelight.com/print-a-semicolon-without-using-semicolon-anywhere-program/)  
[Multiply two numbers without using multiplication operator or loops](http://www.techiedelight.com/multiply-two-numbers-without-using-multiplication-operator-loops/)  
[Find square of a number without using multiplication and division operator | 3 methods](http://www.techiedelight.com/find-square-number-without-using-multiplication-division-operator/)  
[Magnet Puzzle](http://www.techiedelight.com/magnet-puzzle/)