# Array

* 1. [Find pair with given sum in the array](https://www.techiedelight.com/find-pair-with-given-sum-array/)
  2. [Check if subarray with 0 sum is exists or not](https://www.techiedelight.com/check-subarray-with-0-sum-exists-not/)
  3. [Print all sub-arrays with 0 sum](https://www.techiedelight.com/find-sub-array-with-0-sum/)
  4. [Sort binary array in linear time](https://www.techiedelight.com/sort-binary-array-linear-time/)
  5. [Find a duplicate element in a limited range array](https://www.techiedelight.com/find-duplicate-element-limited-range-array/)
  6. [Find maximum length sub-array having given sum](https://www.techiedelight.com/find-maximum-length-sub-array-having-given-sum/)
  7. [Find maximum length sub-array having equal number of 0’s and 1’s](https://www.techiedelight.com/find-maximum-length-sub-array-equal-number-0s-1s/)
  8. [Find maximum product of two integers in an array](https://www.techiedelight.com/find-maximum-product-two-integers-array/)
  9. [Sort an array containing 0’s, 1’s and 2’s (Dutch National Flag Problem)](https://www.techiedelight.com/sort-array-containing-0s-1s-2s-dutch-national-flag-problem/)
  10. [In place merge two sorted arrays](https://www.techiedelight.com/inplace-merge-two-sorted-arrays/)
  11. [Merge two arrays by satisfying given constraints](https://www.techiedelight.com/merge-two-arrays-satisfying-given-constraints/)
  12. [Find index of 0 to replace to get maximum length sequence of continuous ones](https://www.techiedelight.com/find-index-0-replaced-get-maximum-length-sequence-of-continuous-ones/)
  13. [Shuffle a given array of elements (Fisher–Yates shuffle)](https://www.techiedelight.com/shuffle-given-array-elements-fisher-yates-shuffle/)
  14. [Rearrange the array with alternate high and low elements](https://www.techiedelight.com/rearrange-the-array-with-alternate-high-and-low-elements/)
  15. [Find equilibrium index of an array](https://www.techiedelight.com/find-equilibrium-index-array/)
  16. [Find largest sub-array formed by consecutive integers](https://www.techiedelight.com/find-largest-sub-array-formed-by-consecutive-integers/)
  17. [Find majority element (Boyer–Moore Majority Vote Algorithm)](https://www.techiedelight.com/find-majority-element-in-an-array-boyer-moore-majority-vote-algorithm/)
  18. [Move all zeros present in the array to the end](https://www.techiedelight.com/move-zeros-present-array-end/)
  19. [Replace each element of array with product of every other element without using / operator](https://www.techiedelight.com/replace-element-array-product-every-element-without-using-division-operator/)
  20. [Find Longest Bitonic Subarray in an array](https://www.techiedelight.com/find-longest-bitonic-subarray-array/)
  21. [Longest Increasing Subsequence](https://www.techiedelight.com/longest-increasing-subsequence/)
  22. [Find maximum difference between two elements in the array by satisfying given constraints](https://www.techiedelight.com/find-maximum-difference-between-two-elements-array/)
  23. [Maximum Sum Subarray Problem (Kadane’s Algorithm)](https://www.techiedelight.com/maximum-subarray-problem-kadanes-algorithm/)
  24. [Print continuous subarray with maximum sum](https://www.techiedelight.com/print-continuous-subarray-with-maximum-sum/)
  25. [Maximum Sum Circular Subarray](https://www.techiedelight.com/maximum-sum-circular-subarray/)
  26. [Find all distinct combinations of given length — I](https://www.techiedelight.com/find-distinct-combinations-of-given-length/)
  27. [Find all distinct combinations of given length with repetition allowed](https://www.techiedelight.com/find-distinct-combinations-given-length-repetition-allowed/)
  28. [Find maximum sequence of continuous 1’s formed by replacing at-most k zeroes by ones](https://www.techiedelight.com/find-maximum-sequence-of-continuous-1s-can-formed-replacing-k-zeroes-ones/)
  29. [Find minimum sum subarray of given size k](https://www.techiedelight.com/find-minimum-sum-subarray-given-size-k/)
  30. [Find maximum product subarray in a given array](https://www.techiedelight.com/find-maximum-product-subarray-given-array/)
  31. [Find subarray having given sum in given array of integers](https://www.techiedelight.com/find-subarray-having-given-sum-given-array/)
  32. [Find the length of smallest subarray whose sum of elements is greater than the given number](https://www.techiedelight.com/length-of-smallest-subarray-with-sum-greater-number/)
  33. [Find largest number possible from set of given numbers](https://www.techiedelight.com/find-largest-number-possible-set-given-numbers/)
  34. [Find the smallest window in array sorting which will make the entire array sorted](https://www.techiedelight.com/smallest-window-sorting-which-make-array-sorted/)
  35. [Find maximum sum path involving elements of given arrays](https://www.techiedelight.com/find-maximum-sum-path-involving-elements-given-arrays/)
  36. [Maximum profit earned by buying and selling shares any number of times](https://www.techiedelight.com/maximum-profit-earned-buying-and-selling-shares/)
  37. [Trapping Rain Water within given set of bars](https://www.techiedelight.com/trapping-rain-water-within-given-set-bars/)
  38. [Find minimum platforms needed in the station so to avoid any delay in arrival of any train](https://www.techiedelight.com/minimum-number-of-platforms-needed-avoid-delay-arrival-train/)
  39. [Decode the array constructed from another array](https://www.techiedelight.com/decode-the-array-constructed-another-array/)
  40. [Sort an array using one swap](https://www.techiedelight.com/sort-array-using-one-swap/)
  41. [Find Triplet with given sum in an array](https://www.techiedelight.com/find-triplet-given-with-given-sum/)
  42. [Length of longest continuous sequence with same sum in given binary arrays](https://www.techiedelight.com/length-longest-continuous-sequence-same-sum-binary-arrays/)
  43. [Reverse every consecutive m elements of the given subarray](https://www.techiedelight.com/reverse-every-consecutive-m-elements-given-subarray/)
  44. [Maximum Product Subset Problem](https://www.techiedelight.com/maximum-product-subset-problem/)
  45. [Find pairs with given difference k in the array](https://www.techiedelight.com/find-pairs-with-given-difference-array/)
  46. [Find pairs with given difference k in the array | Constant space solution](https://www.techiedelight.com/find-pairs-given-difference-k-array-constant-space-solution/)
  47. [4 sum problem | Quadruplets with given sum](https://www.techiedelight.com/4-sum-problem/)
  48. [Print all quadruplets with given sum | 4-sum problem extended](https://www.techiedelight.com/print-all-quadruplets-with-given-sum-4-sum-problem-extended/)
  49. [Quickselect Algorithm](https://www.techiedelight.com/quickselect-algorithm/)
  50. [Rearrange array such that A[A[i]] is set to i for every element A[i]](https://www.techiedelight.com/rearrange-array-such-that-array-index-is-set-to-i/)
  51. [Print all Triplets that forms Arithmetic Progression](https://www.techiedelight.com/print-triplets-forms-arithmetic-progression/)
  52. [Print all Triplets that forms Geometric Progression](https://www.techiedelight.com/print-triplets-forms-geometric-progression/)
  53. [Print all combination of numbers from 1 to n having sum n](https://www.techiedelight.com/print-all-combination-numbers-from-1-to-n/)
  54. [Replace each element of the array by its corresponding rank in the array](https://www.techiedelight.com/replace-each-element-corresponding-rank-array/)
  55. [Print all Triplets in an array with sum less than or equal to given number](https://www.techiedelight.com/print-triplets-array-sum-less-equal-given-number/)
  56. [Group elements of an array based on their first occurrence](https://www.techiedelight.com/group-elements-array-based-first-occurrence/)
  57. [Find minimum difference between index of two given elements present in the array](https://www.techiedelight.com/find-minimum-difference-index-two-given-elements-present-array/)
  58. [Find maximum absolute difference between sum of two non-overlapping sub-arrays](https://www.techiedelight.com/find-maximum-absolute-difference-subarrays/)
  59. [Find all Symmetric Pairs in an Array of Pairs](https://www.techiedelight.com/find-symmetric-pairs-array-pairs/)
  60. [Partition an array into two sub-arrays with the same sum](https://www.techiedelight.com/partition-array-into-two-sub-arrays-with-same-sum/)
  61. [Find count of distinct elements in every sub-array of size k](https://www.techiedelight.com/count-distinct-elements-every-sub-array-size-k-array/)
  62. [Find two numbers with maximum sum formed by array digits](https://www.techiedelight.com/find-two-numbers-maximum-sum-array-digits/)
  63. [Print all sub-arrays of an array having distinct elements](https://www.techiedelight.com/print-sub-arrays-array-distinct-elements/)
  64. [Find a Triplet having Maximum Product in an Array](https://www.techiedelight.com/find-triplet-maximum-product-array/)
  65. [Find Minimum Index of Repeating Element in an Array](https://www.techiedelight.com/find-minimum-index-repeating-element-array/)
  66. [Generate random input from an array according to given probabilities](https://www.techiedelight.com/generate-random-input-array-according-given-probabilities/)
  67. [Find pair in an array having minimum absolute sum](https://www.techiedelight.com/find-pair-array-minimum-absolute-sum/)
  68. [Find Index of Maximum Occurring Element with Equal Probability](https://www.techiedelight.com/find-index-maximum-occurring-element-equal-probability/)
  69. [Check if an Array is Formed by Consecutive Integers](https://www.techiedelight.com/check-array-formed-consecutive-integers/)
  70. [Find two non-overlapping pairs having same sum in an array](https://www.techiedelight.com/find-two-non-overlapping-pairs-sum-array/)
  71. [Add elements of two arrays into a new array](https://www.techiedelight.com/add-elements-two-arrays/)
  72. [Find Minimum Product among all Combinations of Triplets in an Array](https://www.techiedelight.com/find-minimum-product-triplets-array/)
  73. [Replace every element of an array with the least greater element on its right](https://www.techiedelight.com/replace-every-element-array-least-greater-element-right/)
  74. [Find all odd occurring elements in an array having limited range of elements](https://www.techiedelight.com/find-odd-occurring-elements-array/)
  75. [Count the distinct absolute values in the sorted array](https://www.techiedelight.com/count-distinct-absolute-values-sorted-array/)
  76. [Print all combinations of positive integers in increasing order that sum to a given number](https://www.techiedelight.com/print-combinations-integers-sum-given-number/)
  77. [Find all distinct combinations of given length — II](https://www.techiedelight.com/find-distinct-combinations-given-length-2/)
  78. [Find subarrays with given sum in an array](https://www.techiedelight.com/find-subarrays-given-sum-array/)
  79. [Find the surpasser count for each element of an array](https://www.techiedelight.com/surpasser-count-each-element-array/)
  80. [Find maximum length sequence of continuous ones (Using Sliding Window)](https://www.techiedelight.com/find-maximum-length-sequence-continuous-ones-sliding-window/)
  81. [Find maximum length sequence of continuous ones](https://www.techiedelight.com/find-maximum-length-sequence-continuous-ones/)
  82. [Find index that divides an array into two non-empty subarrays of equal sum](https://www.techiedelight.com/find-index-that-divides-array-non-empty-subarrays-equal-sum/)
  83. [Calculate frequency of all elements present in an array of specified range](https://www.techiedelight.com/calculate-frequency-elements-present-array-specified-range/)
  84. [Rearrange the array such that it contains positive and negative numbers at alternate positions](https://www.techiedelight.com/rearrange-array-positive-negative-numbers-alternate-positions/)
  85. [Find a sorted triplet in the given array](https://www.techiedelight.com/find-sorted-triplet-array/)
  86. [Shuffle an array according to the given order of elements](https://www.techiedelight.com/shuffle-array-according-to-given-order/)
  87. [Count number of strictly increasing sub-arrays in an array](https://www.techiedelight.com/count-strictly-increasing-sub-arrays/)
  88. [Find duplicates within given range k in an array](https://www.techiedelight.com/find-duplicates-within-given-range-array/)
  89. [Longest Alternating Subarray Problem](https://www.techiedelight.com/longest-alternating-subarray-problem/)
  90. [Find minimum range with at-least one element from each of the given arrays](https://www.techiedelight.com/find-minimum-range-given-arrays/)
  91. [Find longest subsequence formed by consecutive integers](https://www.techiedelight.com/find-longest-subsequence-formed-by-consecutive-integers/)
  92. [Find all elements in an array that are greater than all elements present to their right](https://www.techiedelight.com/find-elements-array-greater-than-elements-right/)
  93. [Find missing number in array without using extra space](https://www.techiedelight.com/find-missing-number-array-without-extra-space/)
  94. [Determine index of an element in given array which satisfies given constraints](https://www.techiedelight.com/determine-index-element-array/)
  95. [Find minimum moves required for converting a given array to an array of zeroes](https://www.techiedelight.com/find-minimum-moves-required-converting-array/)
  96. [Left rotate an array](https://www.techiedelight.com/left-rotate-array-c/)
  97. [Right rotate an array k times](https://www.techiedelight.com/right-rotate-an-array-k-times/)
  98. [Find maximum profit earned from at most two stock transactions](https://www.techiedelight.com/find-maximum-profit-earned-from-at-most-two-stock-transactions/)
  99. [Find Frequency of each element in a sorted array containing duplicates](https://www.techiedelight.com/find-frequency-element-sorted-array-containing-duplicates/)
  100. [Find Minimum and Maximum element in an array using minimum comparisons](https://www.techiedelight.com/find-minimum-maximum-element-array-using-minimum-comparisons/)
  101. [Difference between Subarray, Subsequence and Subset](https://www.techiedelight.com/difference-between-subarray-subsequence-subset/)
  102. [Find odd occurring element in an array in single traversal](https://www.techiedelight.com/find-odd-occurring-element-array-single-traversal/)
  103. [Find odd occurring element in logarithmic time](https://www.techiedelight.com/find-odd-occurring-element-logn-time/)
  104. [Find two odd occurring elements in an array without using any extra space](https://www.techiedelight.com/find-two-odd-occurring-element-array-without-extra-space/)
  105. [Check if given array represents min heap or not](https://www.techiedelight.com/check-given-array-represents-min-heap-not/)
  106. [Find K’th smallest element in an array](https://www.techiedelight.com/find-kth-smallest-element-array/)
  107. [Find K’th largest element in an array](https://www.techiedelight.com/find-kth-largest-element-array/)
  108. [Sort a K-Sorted Array](https://www.techiedelight.com/sort-k-sorted-array/)
  109. [Merge M sorted lists of variable length](https://www.techiedelight.com/merge-m-sorted-lists-variable-length/)
  110. [Find smallest range with at-least one element from each of the given lists](https://www.techiedelight.com/find-smallest-range-least-one-element-given-lists/)
  111. [Merge M sorted lists each containing N elements](https://www.techiedelight.com/merge-m-sorted-lists-containing-n-elements/)
  112. [Find maximum sum of subsequence with no adjacent elements](https://www.techiedelight.com/maximum-sum-of-subsequence-with-no-adjacent-elements/)
  113. [Find ways to calculate a target from elements of specified array](https://www.techiedelight.com/find-ways-calculate-target-elements-array/)
  114. [Sort elements by their frequency and Index](https://www.techiedelight.com/sort-elements-by-their-frequency-and-index/)
  115. [Sort an array based on order defined by another array](https://www.techiedelight.com/custom-sort-sort-elements-array-order-elements-defined-second-array/)
  116. [Inversion Count of an array](https://www.techiedelight.com/inversion-count-array/)
  117. [Segregate positive and negative integers in linear time](https://www.techiedelight.com/positive-and-negative-integers-segregate/)
  118. [Find number of rotations in a circularly sorted array](https://www.techiedelight.com/find-number-rotations-circularly-sorted-array/)
  119. [Search an element in a circular sorted array](https://www.techiedelight.com/search-element-circular-sorted-array/)
  120. [Find first or last occurrence of a given number in a sorted array](https://www.techiedelight.com/find-first-or-last-occurrence-of-a-given-number-sorted-array/)
  121. [Count occurrences of a number in a sorted array with duplicates](https://www.techiedelight.com/count-occurrences-number-sorted-array-duplicates/)
  122. [Find smallest missing element from a sorted array](https://www.techiedelight.com/find-smallest-missing-element-sorted-array/)
  123. [Find Floor and Ceil of a number in a sorted array](https://www.techiedelight.com/find-floor-ceil-number-sorted-array/)
  124. [Search in a nearly sorted array in logarithmic time](https://www.techiedelight.com/search-nearly-sorted-array-ologn-time/)
  125. [Find number of 1’s in a sorted binary array](https://www.techiedelight.com/find-number-1s-sorted-binary-array/)
  126. [Find Missing Term in a Sequence in Logarithmic time](https://www.techiedelight.com/find-missing-term-sequence-ologn-time/)
  127. [Find missing number and duplicate elements in an array](https://www.techiedelight.com/find-missing-number-duplicate-elements-array/)
  128. [Find the peak element in an array](https://www.techiedelight.com/find-peak-element-array/)
  129. [Find Floor and Ceil of a number in a sorted array (Recursive solution)](https://www.techiedelight.com/find-floor-ceil-number-sorted-array-recursive/)
  130. [Print all distinct subsets of a given set](https://www.techiedelight.com/print-distinct-subsets-given-set/)
  131. [Find two duplicate elements in a limited range array (using XOR)](https://www.techiedelight.com/find-two-duplicate-elements-limited-range-array-using-xor/)
  132. [Combinations of words formed by replacing given numbers with corresponding alphabets](https://www.techiedelight.com/combinations-of-words-formed-replacing-given-numbers-corresponding-english-alphabet/)
  133. [0–1 Knapsack Problem](https://www.techiedelight.com/0-1-knapsack-problem/)
  134. [Subset sum Problem](https://www.techiedelight.com/subset-sum-problem/)
  135. [Partition Problem](https://www.techiedelight.com/partition-problem/)
  136. [3-Partition Problem](https://www.techiedelight.com/3-partition-problem/)
  137. [3-partition problem extended | Print all partitions](https://www.techiedelight.com/3-partition-problem-extended-print-all-partitions/)
  138. [K-Partition Problem | Printing all Partitions](https://www.techiedelight.com/k-partition-problem-print-all-subsets/)
  139. [Minimum Sum Partition Problem](https://www.techiedelight.com/minimum-sum-partition-problem/)
  140. [Rod Cutting](https://www.techiedelight.com/rot-cutting/)
  141. [Longest Alternating Subsequence Problem](https://www.techiedelight.com/longest-alternating-subsequence/)
  142. [Coin change-making problem (unlimited supply of coins)](https://www.techiedelight.com/coin-change-making-problem-unlimited-supply-coins/)
  143. [Coin Change Problem — Find total number of ways to get the denomination of coins](https://www.techiedelight.com/coin-change-problem-find-total-number-ways-get-denomination-coins/)
  144. [Find maximum profit earned from at most K stock transactions](https://www.techiedelight.com/find-maximum-profit-earned-at-most-k-stock-transactions/)

# Stack

* 1. Stack Implementation using Array/List — [C](https://www.techiedelight.com/stack-implementation/), [C++](https://www.techiedelight.com/stack-implementation-in-cpp/), [Java](https://www.techiedelight.com/stack-implementation-in-java/), [Python](https://www.techiedelight.com/stack-implementation-python/)
  2. [Stack Implementation using Linked List](https://www.techiedelight.com/stack-implementation-using-linked-list/)
  3. [Check if given expression is balanced expression or not](https://www.techiedelight.com/check-given-expression-balanced-expression-not/)
  4. [Find duplicate parenthesis in an expression](https://www.techiedelight.com/find-duplicate-parenthesis-expression/)
  5. [Evaluate given postfix expression](https://www.techiedelight.com/evaluate-given-postfix-expression/)
  6. [Decode the given sequence to construct minimum number without repeated digits](https://www.techiedelight.com/decode-the-given-sequence-construct-minimum-number-without-repeated-digits/)
  7. [Design a stack which returns minimum element in constant time](https://www.techiedelight.com/design-stack-which-returns-minimum-element-constant-time/)
  8. [Design a stack which returns minimum element without using auxiliary stack](https://www.techiedelight.com/design-a-stack-which-returns-minimum-element-without-using-auxiliary-stack/)
  9. [Merging Overlapping Intervals](https://www.techiedelight.com/merging-overlapping-intervals/)
  10. [Reverse String without using Recursion](https://www.techiedelight.com/reverse-string-without-using-recursion/)
  11. [Implement Stack using Queue Data Structure](https://www.techiedelight.com/implement-stack-using-queue-data-structure/)
  12. [Implement a Queue using Stack Data Structure](https://www.techiedelight.com/implement-a-queue-using-stack-data-structure/)
  13. [Implement two stacks in a single array](https://www.techiedelight.com/implement-two-stacks-single-array/)
  14. [Recursive solution to sort a stack](https://www.techiedelight.com/recursive-solution-sort-stack/)
  15. [Find length of the longest balanced parenthesis in a string](https://www.techiedelight.com/find-length-longest-balanced-parenthesis-string/)
  16. [Reverse a string using stack data structure](https://www.techiedelight.com/reverse-a-string-using-stack-data-structure/)
  17. [Find all elements in an array that are greater than all elements present to their right](https://www.techiedelight.com/find-elements-array-greater-than-elements-right/)
  18. [Inorder Tree Traversal](https://www.techiedelight.com/inorder-tree-traversal-iterative-recursive/)
  19. [Preorder Tree Traversal](https://www.techiedelight.com/preorder-tree-traversal-iterative-recursive/)
  20. [Postorder Tree Traversal](https://www.techiedelight.com/postorder-tree-traversal-iterative-recursive/)
  21. [Find preorder traversal of a binary tree from its inorder and postorder sequence](https://www.techiedelight.com/find-preorder-traversal-binary-tree-from-inorder-postorder/)
  22. [Find ancestors of given node in a Binary Tree](https://www.techiedelight.com/find-ancestors-of-given-node-binary-tree/)
  23. [Check if two given binary trees are identical or not](https://www.techiedelight.com/check-if-two-binary-trees-are-identical-not-iterative-recursive/)
  24. [Reverse Level Order Traversal of Binary Tree](https://www.techiedelight.com/reverse-level-order-traversal-binary-tree/)
  25. [Reverse given text without reversing the individual words](https://www.techiedelight.com/reverse-text-without-reversing-individual-words/)
  26. [Find all binary strings that can be formed from given wildcard pattern](https://www.techiedelight.com/find-binary-strings-can-formed-given-wildcard-pattern/)
  27. [Iterative Implementation of QuickSort](https://www.techiedelight.com/iterative-implementation-of-quicksort/)
  28. [Depth First Search (DFS) Algorithm](https://www.techiedelight.com/depth-first-search/)
  29. [Invert a Binary Tree](https://www.techiedelight.com/invert-binary-tree-recursive-iterative/)
  30. [Print leaf to root path for every leaf node in a binary tree](https://www.techiedelight.com/print-leaf-to-root-path-binary-tree/)
  31. [Longest Increasing Subsequence](https://www.techiedelight.com/longest-increasing-subsequence/)
  32. [Invert alternate levels of a perfect binary tree](https://www.techiedelight.com/invert-alternate-levels-perfect-binary-tree/)

# Queue

* 1. Queue Implementation using Array/List — [C](https://www.techiedelight.com/circular-queue-implementation-c/), [C++](https://www.techiedelight.com/queue-implementation-cpp/), [Java](https://www.techiedelight.com/queue-implementation-in-java/), [Python](https://www.techiedelight.com/queue-implementation-python/)
  2. [Queue Implementation using Linked List](https://www.techiedelight.com/queue-implementation-using-linked-list/)
  3. [Implement Stack using Queue Data Structure](https://www.techiedelight.com/implement-stack-using-queue-data-structure/)
  4. [Implement a Queue using Stack Data Structure](https://www.techiedelight.com/implement-a-queue-using-stack-data-structure/)
  5. [Efficiently print all nodes between two given levels in a binary tree](https://www.techiedelight.com/print-nodes-between-two-levels-binary-tree/)
  6. [Chess Knight Problem — Find Shortest path from source to destination](https://www.techiedelight.com/chess-knight-problem-find-shortest-path-source-destination/)
  7. [Shortest path in a Maze | Lee Algorithm](https://www.techiedelight.com/lee-algorithm-shortest-path-in-a-maze/)
  8. [Find shortest safe route in a field with sensors present](https://www.techiedelight.com/find-shortest-safe-route-field-sensors-present/)
  9. [Flood Fill Algorithm](https://www.techiedelight.com/flood-fill-algorithm/)
  10. [Count the number of islands](https://www.techiedelight.com/count-the-number-of-islands/)
  11. [Find Shortest path from source to destination in a matrix that satisfies given constraints](https://www.techiedelight.com/find-shortest-path-source-destination-matrix-satisfies-given-constraints/)
  12. [Generate binary numbers between 1 to N](https://www.techiedelight.com/generate-binary-numbers-1-n/)
  13. [Calculate height of a binary tree](https://www.techiedelight.com/calculate-height-binary-tree-iterative-recursive/)
  14. [Delete given Binary Tree](https://www.techiedelight.com/delete-given-binary-tree-iterative-recursive/)
  15. [Level Order Traversal of Binary Tree](https://www.techiedelight.com/level-order-traversal-binary-tree/)
  16. [Spiral Order Traversal of Binary Tree](https://www.techiedelight.com/spiral-order-traversal-binary-tree/)
  17. [Reverse Level Order Traversal of Binary Tree](https://www.techiedelight.com/reverse-level-order-traversal-binary-tree/)
  18. [Print all nodes of a given binary tree in specific order](https://www.techiedelight.com/print-nodes-binary-tree-specific-order/)
  19. [Print left view of binary tree](https://www.techiedelight.com/print-left-view-of-binary-tree/)
  20. [Find next node in same level for given node in a binary tree](https://www.techiedelight.com/find-next-node-in-same-level-binary-tree/)
  21. [Check if given binary tree is complete binary tree or not](https://www.techiedelight.com/check-given-binary-tree-complete-binary-tree-not/)
  22. [Print Diagonal Traversal of Binary Tree](https://www.techiedelight.com/print-diagonal-traversal-binary-tree/)
  23. [Print corner nodes of every level in binary tree](https://www.techiedelight.com/print-corner-nodes-every-level-binary-tree/)
  24. [Invert a Binary Tree](https://www.techiedelight.com/invert-binary-tree-recursive-iterative/)
  25. [Find minimum passes required to convert all negative values in the matrix](https://www.techiedelight.com/find-minimum-passes-required-convert-negative-values-matrix/)
  26. [Convert a Binary Tree into a Doubly Linked List in Spiral Order](https://www.techiedelight.com/convert-binary-tree-into-doubly-linked-list/)
  27. [Check if a binary tree is a min-heap or not](https://www.techiedelight.com/check-binary-tree-is-min-heap/)
  28. [Invert alternate levels of a perfect binary tree](https://www.techiedelight.com/invert-alternate-levels-perfect-binary-tree/)
  29. [Convert a Binary Search Tree into a Min Heap](https://www.techiedelight.com/convert-binary-search-tree-into-min-heap/)
  30. [Snake and Ladder Problem](https://www.techiedelight.com/min-throws-required-to-win-snake-and-ladder-game/)
  31. [Find shortest distance of every cell from landmine in a Maze](https://www.techiedelight.com/find-shortest-distance-every-cell-landmine-maze/)
  32. [Convert a multilevel linked list to a singly linked list](https://www.techiedelight.com/convert-multilevel-linked-list-singly/)
  33. [Breadth First Search (BFS) Algorithm](https://www.techiedelight.com/breadth-first-search/)
  34. [Check if an undirected graph contains cycle or not](https://www.techiedelight.com/check-undirected-graph-contains-cycle-not/)
  35. [Find maximum cost path in graph from given source to destination](https://www.techiedelight.com/maximum-cost-path-graph-source-destination/)
  36. [Total paths in given digraph from given source to destination having exactly m edges](https://www.techiedelight.com/total-paths-in-digraph-from-source-to-destination-m-edges/)
  37. [Least cost path in given digraph from given source to destination having exactly m edges](https://www.techiedelight.com/least-cost-path-digraph-source-destination-m-edges/)

# String

* 1. [Check if given string is a rotated palindrome or not](https://www.techiedelight.com/check-given-string-rotated-palindrome-not/)
  2. [Longest Palindromic Substring (Non-DP Space Optimized Solution)](https://www.techiedelight.com/longest-palindromic-substring-non-dp-space-optimized-solution/)
  3. [Check if repeated subsequence is present in the string or not](https://www.techiedelight.com/check-repeated-subsequence-present-string-not/)
  4. [Check if strings can be derived from each other by circularly rotating them](https://www.techiedelight.com/check-strings-can-derived-circularly-rotating/)
  5. [Check if given set of moves is circular or not](https://www.techiedelight.com/check-given-set-moves-circular-not/)
  6. [Convert given number into corresponding excel column name](https://www.techiedelight.com/convert-given-number-corresponding-excel-column-name/)
  7. [Determine if two strings are anagram or not](https://www.techiedelight.com/determine-if-two-strings-are-anagram-or-not/)
  8. [Find all binary strings that can be formed from given wildcard pattern](https://www.techiedelight.com/find-binary-strings-can-formed-given-wildcard-pattern/)
  9. [Find all interleaving of given strings](https://www.techiedelight.com/find-interleavings-of-given-strings/)
  10. [Isomorphic Strings](https://www.techiedelight.com/isomorphic-strings/)
  11. [Find all possible palindromic substrings in a string](https://www.techiedelight.com/find-possible-palindromic-substrings-string/)
  12. [Find all possible combinations of words formed from mobile keypad](https://www.techiedelight.com/find-possible-combinations-words-formed-from-mobile-keypad/)
  13. [Find all possible combinations by replacing given digits with characters of the corresponding list](https://www.techiedelight.com/possible-combinations-replacing-given-digits-corresponding-list/)
  14. [Find all words from given list that follows same order of characters as given pattern](https://www.techiedelight.com/find-words-that-follows-given-pattern/)
  15. [Group anagrams together from given list of words](https://www.techiedelight.com/group-anagrams-together-given-list-words/)
  16. [Find minimum operations required to transform a string into another string](https://www.techiedelight.com/find-minimum-operations-required-transform-string-into-another/)
  17. [Determine if a string can be transformed into another string with a single edit](https://www.techiedelight.com/determine-string-transformed-into-another-string-single-edit/)
  18. [Find length of the longest balanced parenthesis in a string](https://www.techiedelight.com/find-length-longest-balanced-parenthesis-string/)
  19. [In place remove all occurrences of ‘AB’ and ‘C’ from the string](https://www.techiedelight.com/inplace-remove-all-occurrences-ab-c-string/)
  20. [Longest even length palindromic sum substring](https://www.techiedelight.com/longest-even-length-palidromic-sum-substring/)
  21. [Print string in zig-zag form in k rows](https://www.techiedelight.com/print-string-in-zig-zag-form-k-rows/)
  22. [Reverse given text without reversing the individual words](https://www.techiedelight.com/reverse-text-without-reversing-individual-words/)
  23. [Run Length Encoding (RLE) Data Compression Algorithm](https://www.techiedelight.com/run-length-encoding-rle-data-compression-algorithm/)
  24. [Find the longest substring of given string containing k distinct characters](https://www.techiedelight.com/find-longest-substring-containing-k-distinct-characters/)
  25. [Find all palindromic permutations of a string](https://www.techiedelight.com/find-palindromic-permutations-string/)
  26. [Find all substrings of a string that are permutation of a given string](https://www.techiedelight.com/find-substrings-string-permutation-given-string/)
  27. [Find the longest substring of given string containing all distinct characters](https://www.techiedelight.com/find-longest-substring-given-string-containing-distinct-characters/)
  28. [Iterative Approach to find Permutations of a String](https://www.techiedelight.com/find-permutations-string-cpp-java-iterative/)
  29. [Generate all Permutations of a String in Java](https://www.techiedelight.com/generate-permutations-string-java-recursive-iterative/)
  30. [Find all lexicographically next permutations of a string sorted in ascending order](https://www.techiedelight.com/find-lexicographically-next-permutations-string-sorted-ascending-order/)
  31. [Find Lexicographically minimal string rotation](https://www.techiedelight.com/find-lexicographically-minimal-string-rotation/)
  32. [Find all strings of given length containing balanced parentheses](https://www.techiedelight.com/find-strings-given-length-containing-balanced-parentheses/)
  33. [Find all combinations of non-overlapping substrings of a string](https://www.techiedelight.com/find-combinations-non-overlapping-substrings-string/)
  34. [Determine if a given string is palindrome or not](https://www.techiedelight.com/determine-given-string-is-palindrome-not/)
  35. [Find the minimum number of inversions needed to make the given expression balanced](https://www.techiedelight.com/minimum-number-inversions-expression-balanced/)
  36. [Construct the longest palindrome by shuffling or deleting characters from a string](https://www.techiedelight.com/construct-longest-palindrome-string/)
  37. [Print all combinations of phrases formed by picking words from each of the given lists](https://www.techiedelight.com/combinations-phrases-formed-picking-words-lists/)
  38. [Break a string into all possible combinations of non-overlapping substrings](https://www.techiedelight.com/break-string-non-overlapping-substrings/)
  39. [Remove all extra spaces from a string](https://www.techiedelight.com/remove-extra-spaces-string/)
  40. [Remove adjacent duplicate characters from a string](https://www.techiedelight.com/remove-adjacent-duplicates-characters-string/)
  41. [Find first non-repeating character in a string by doing only one traversal of it](https://www.techiedelight.com/find-first-non-repeating-character-string-one-traversal/)
  42. [Find all N-digit strictly increasing numbers (Bottom-Up and Top-Down Approach)](https://www.techiedelight.com/find-n-digit-strictly-increasing-numbers-bottom-top-approach/)
  43. [Find all N-digit binary numbers having more 1’s than 0’s for any prefix](https://www.techiedelight.com/find-n-digit-binary-numbers-having-more-one-than-zero/)
  44. [Find all N-digit numbers with given sum of digits](https://www.techiedelight.com/find-all-n-digit-numbers-given-sum-digits/)
  45. [Find all N-digit binary numbers with k-bits set where k ranges from 1 to N](https://www.techiedelight.com/find-n-digit-binary-numbers-k-bits-set-k-ranges-1-n/)
  46. [Find all N-digit binary numbers with equal sum of bits in its two halves](https://www.techiedelight.com/find-n-digit-binary-numbers-equal-sum-bits-two-halves/)
  47. [Find all N-digit numbers with equal sum of digits at even and odd index](https://www.techiedelight.com/find-n-digit-numbers-with-equal-sum-even-odd-digits/)
  48. [Find all Lexicographic Permutations of a String](https://www.techiedelight.com/find-lexicographic-permutations-string/)
  49. [Lexicographic Rank of a String](https://www.techiedelight.com/calculate-rank-lexicographically-sorted-permutations/)
  50. [Find all lexicographically previous permutations of a string sorted in descending order](https://www.techiedelight.com/find-all-lexicographically-previous-permutations-string/)
  51. [Replace all non-overlapping occurrences of the pattern](https://www.techiedelight.com/replace-non-overlapping-occurrences-pattern/)
  52. [Introduction to Pattern Matching](https://www.techiedelight.com/introduction-pattern-matching/)
  53. [Implementation of KMP Algorithm](https://www.techiedelight.com/implementation-kmp-algorithm-c-cpp-java/)
  54. [Reverse String without using Recursion](https://www.techiedelight.com/reverse-string-without-using-recursion/)
  55. [Reverse given string using Recursion](https://www.techiedelight.com/reverse-a-string-using-recursion/)
  56. [Determine if characters of a String follow a specified order or not](https://www.techiedelight.com/determine-string-follows-specified-order/)
  57. [In-place remove all adjacent duplicates from the given string](https://www.techiedelight.com/in-place-remove-all-adjacent-duplicates-from-string/)
  58. [Check if given sentence is syntactically correct or not](https://www.techiedelight.com/check-given-sentence-syntactically-correct-not/)
  59. [Find all Permutations of a given string](https://www.techiedelight.com/find-permutations-given-string/)
  60. [Find first k non-repeating characters in a string in single traversal](https://www.techiedelight.com/first-k-non-repeating-characters-string/)
  61. [Check if given string is interleaving of two other given strings](https://www.techiedelight.com/check-string-interleaving-two-given-strings/)
  62. [Decode the given sequence to construct minimum number without repeated digits](https://www.techiedelight.com/decode-the-given-sequence-construct-minimum-number-without-repeated-digits/)
  63. [Combinations of words formed by replacing given numbers with corresponding alphabets](https://www.techiedelight.com/combinations-of-words-formed-replacing-given-numbers-corresponding-english-alphabet/)
  64. [Count number of times a pattern appears in given string as a subsequence](https://www.techiedelight.com/count-number-times-pattern-appears-given-string-subsequence/)
  65. [Check if a string matches with a given wildcard pattern](https://www.techiedelight.com/check-string-matches-with-wildcard-pattern/)
  66. [Find all words matching a pattern in the given dictionary](https://www.techiedelight.com/find-all-words-matching-pattern-dictionary/)
  67. [The Levenshtein Distance (Edit Distance) Problem](https://www.techiedelight.com/levenshtein-distance-edit-distance-problem/)
  68. [Longest Common Subsequence Problem](https://www.techiedelight.com/longest-common-subsequence/)
  69. [Longest Repeated Subsequence Problem](https://www.techiedelight.com/longest-repeated-subsequence-problem/)
  70. [Longest Palindromic Subsequence using Dynamic Programming](https://www.techiedelight.com/longest-palindromic-subsequence-using-dynamic-programming/)
  71. [Longest Common Substring Problem](https://www.techiedelight.com/longest-common-substring-problem/)
  72. [Shortest Common Supersequence Problem](https://www.techiedelight.com/shortest-common-supersequence-introduction-scs-length/)
  73. [Word Break Problem](https://www.techiedelight.com/word-break-problem/)
  74. [Wildcard Pattern Matching](https://www.techiedelight.com/wildcard-pattern-matching/)
  75. [Find minimum cuts needed for palindromic partition of a string](https://www.techiedelight.com/find-minimum-cuts-needed-palindromic-partition-string/)
  76. [Check if a string is K-Palindrome or not](https://www.techiedelight.com/check-given-string-k-palindrome-not/)
  77. [Find shortest route in a device to construct the given string](https://www.techiedelight.com/find-shortest-route-device-construct-given-string/)
  78. [Find minimum number possible by doing at-most K swaps](https://www.techiedelight.com/find-minimum-number-possible-k-swaps/)
  79. [Determine if a pattern matches with a string or not](https://www.techiedelight.com/determine-pattern-matches-string-not/)
  80. [Find minimum number of deletions required to convert a string into palindrome](https://www.techiedelight.com/find-minimum-number-deletions-convert-string-into-palindrome/)

# Linked List

* 1. [Introduction to Linked Lists](https://www.techiedelight.com/introduction-linked-lists/)
  2. Linked List Implementation — [C](https://www.techiedelight.com/linked-list-implementation-part-1/), [C++](https://www.techiedelight.com/linked-list-implementation-cpp/), [Java](https://www.techiedelight.com/linked-list-implementation-java/), [Python](https://www.techiedelight.com/linked-list-implementation-python/)
  3. [Linked List | Insertion at Tail](https://www.techiedelight.com/linked-list-implementation-part-2/)
  4. [Static Linked List](https://www.techiedelight.com/static-linked-list-c/)
  5. [Clone given Linked List](https://www.techiedelight.com/clone-given-linked-list/)
  6. [Delete Linked List](https://www.techiedelight.com/delete-linked-list/)
  7. [Pop operation in linked list](https://www.techiedelight.com/pop-operation-in-linked-list/)
  8. [Insert given node into the correct sorted position in the given sorted linked list](https://www.techiedelight.com/sorted-insert-in-linked-list/)
  9. [Rearrange linked list in increasing order (Sort linked list)](https://www.techiedelight.com/given-linked-list-change-sorted-order/)
  10. [Split the nodes of the given linked list into front and back halves](https://www.techiedelight.com/split-nodes-given-linked-list-front-back-halves/)
  11. [Remove duplicates from a sorted linked list](https://www.techiedelight.com/remove-duplicates-sorted-linked-list/)
  12. [Move front node of the given list to the front of the another list](https://www.techiedelight.com/move-front-node-given-list-front-another-list/)
  13. [Move even nodes to the end of the list in reverse order](https://www.techiedelight.com/move-even-nodes-to-end-of-list-in-reverse-order/)
  14. [Split given linked list into two lists where each list containing alternating elements from it](https://www.techiedelight.com/split-linked-list-into-two-lists-list-containing-alternating-elements/)
  15. [Construct a linked list by merging alternate nodes of two given lists](https://www.techiedelight.com/merge-alternate-nodes-two-linked-lists/)
  16. [Merge Sort Algorithm for Singly Linked List](https://www.techiedelight.com/merge-sort-singly-linked-list/)
  17. [Merge two sorted linked lists into one](https://www.techiedelight.com/merge-given-sorted-linked-lists/)
  18. [Merge K sorted linked lists](https://www.techiedelight.com/efficiently-merge-k-sorted-linked-lists/)
  19. [Intersection of two given sorted linked lists](https://www.techiedelight.com/intersection-two-given-sorted-linked-lists/)
  20. [Reverse Linked List (Iterative Solution)](https://www.techiedelight.com/reverse-linked-list-part-1-iterative-solution/)
  21. [Reverse Linked List (Recursive Solution)](https://www.techiedelight.com/reverse-linked-list-part-2-recursive-solution/)
  22. [Reverse every group of k nodes in given linked list](https://www.techiedelight.com/reverse-every-k-nodes-of-a-linked-list/)
  23. [Find K’th node from the end in a linked list](https://www.techiedelight.com/find-kth-node-from-the-end-linked-list/)
  24. [Merge alternate nodes of two linked lists into the first list](https://www.techiedelight.com/merge-alternate-nodes-two-linked-lists-first-list/)
  25. [Merge two sorted linked lists from their end](https://www.techiedelight.com/merge-two-sorted-linked-lists-end/)
  26. [Delete every N nodes in a linked list after skipping M nodes](https://www.techiedelight.com/delete-every-n-nodes-linked-list-skipping-m-nodes/)
  27. [Rearrange linked list in specific manner in linear time](https://www.techiedelight.com/rearrange-linked-list-specific-manner-linear-time/)
  28. [Check if linked list is palindrome or not](https://www.techiedelight.com/check-if-linked-list-is-palindrome/)
  29. [Move last node to front in a given Linked List](https://www.techiedelight.com/move-last-node-to-front-linked-list/)
  30. [Rearrange the linked list in specific manner](https://www.techiedelight.com/rearrange-the-linked-list-specific-manner/)
  31. [Detect Cycle in a linked list (Floyd’s Cycle Detection Algorithm)](https://www.techiedelight.com/detect-cycle-linked-list-floyds-cycle-detection-algorithm/)
  32. [Sort linked list containing 0’s, 1’s and 2’s](https://www.techiedelight.com/sort-linked-list-containing-0s-1s-2s/)
  33. [Implement Stack using Linked List](https://www.techiedelight.com/stack-implementation-using-linked-list/)
  34. [Implement Queue using Linked List](https://www.techiedelight.com/queue-implementation-using-linked-list/)
  35. [Remove duplicates from a linked list](https://www.techiedelight.com/remove-duplicates-linked-list/)
  36. [Rearrange the linked list so that it has alternating high, low values](https://www.techiedelight.com/rearrange-linked-list-alternating-high-low-values/)
  37. [Rearrange a Linked List by Separating Odd Nodes from the Even Ones](https://www.techiedelight.com/rearrange-linked-list-separating-odd-nodes-even/)
  38. [Calculate height of a binary tree with leaf nodes forming a circular doubly linked list](https://www.techiedelight.com/calculate-height-binary-tree-leaf-nodes-forming-circular-doubly-linked-list/)
  39. [XOR Linked List: Overview and Implementation](https://www.techiedelight.com/xor-linked-list-overview-implementation-c-cpp/)
  40. [Convert a multilevel linked list to a singly linked list](https://www.techiedelight.com/convert-multilevel-linked-list-singly/)
  41. [Recursively check if linked list of characters is palindrome or not](https://www.techiedelight.com/recursively-check-linked-list-characters-palindrome-or-not/)
  42. [Merge two BSTs into a doubly linked list in sorted order](https://www.techiedelight.com/merge-two-bsts-into-doubly-linked-list-sorted-order/)
  43. [Remove redundant nodes from a path formed by a linked list](https://www.techiedelight.com/remove-redundant-nodes-path-formed-linked-list/)
  44. [Add a single-digit number to a linked list representing a number](https://www.techiedelight.com/add-single-digit-number-linked-list-representing-number/)
  45. [Reverse every alternate group of k nodes in a linked list](https://www.techiedelight.com/reverse-alternate-group-k-nodes-linked-list/)
  46. [Determine if a given linked list is a palindrome or not](https://www.techiedelight.com/determine-linked-list-palindrome-or-not/)
  47. [Sort a Doubly Linked List using Merge Sort](https://www.techiedelight.com/sort-doubly-linked-list-merge-sort/)
  48. [Reverse a Doubly Linked List](https://www.techiedelight.com/reverse-doubly-linked-list/)
  49. [Pairwise swap adjacent nodes of a linked list](https://www.techiedelight.com/pairwise-swap-adjacent-nodes-linked-list/)
  50. [Flatten a linked list](https://www.techiedelight.com/flatten-linked-list/)
  51. [Check if a Linked List of String is Palindromic](https://www.techiedelight.com/check-linked-list-strings-palindromic/)
  52. [Flatten a multilevel linked list](https://www.techiedelight.com/flatten-multilevel-linked-list/)
  53. [Construct a height-balanced BST from an unbalanced BST](https://www.techiedelight.com/construct-height-balanced-bst-from-unbalanced-bst/)
  54. [Swap K’th node from beginning with K’th node from end in a Linked List](https://www.techiedelight.com/swap-kth-node-beginning-with-kth-node-end/)
  55. [Add two linked lists without using any extra space](https://www.techiedelight.com/add-two-linked-lists-without-using-extra-space/)
  56. [Clone a Linked List with Random Pointers](https://www.techiedelight.com/clone-a-linked-list-with-random-pointers/)
  57. [Update random pointer for each linked list node to point to the maximum node](https://www.techiedelight.com/update-random-pointer-linked-list-node-maximum-node/)
  58. [Link nodes present in each level of a binary tree in the form of a linked list](https://www.techiedelight.com/link-nodes-each-level-binary-tree/)
  59. [Convert a Ternary Tree to a Doubly Linked List](https://www.techiedelight.com/convert-ternary-tree-doubly-linked-list/)
  60. [Print nodes of a given binary tree in vertical order](https://www.techiedelight.com/print-nodes-binary-tree-vertical-order/)
  61. [Convert a Binary Tree into a Doubly Linked List in Spiral Order](https://www.techiedelight.com/convert-binary-tree-into-doubly-linked-list/)
  62. [Construct a Height-Balanced BST from a Sorted Doubly Linked List](https://www.techiedelight.com/construct-height-balanced-bst-from-sorted-doubly-linked-list/)
  63. [In-place merge two sorted linked lists without modifying links of the first list](https://www.techiedelight.com/in-place-merge-two-sorted-linked-lists/)
  64. [Reverse specified portion of a Linked List](https://www.techiedelight.com/reverse-specified-portion-linked-list/)

# Binary Bitwise

* 1. [Bit Hacks — Part 1 (Basic)](https://www.techiedelight.com/bit-hacks-part-1-basic/)
  2. [Bit Hacks — Part 2 (Playing with k’th bit)](https://www.techiedelight.com/bit-hacks-part-2-playing-kth-bit/)
  3. [Bit Hacks — Part 3 (Playing with rightmost set bit of a number)](https://www.techiedelight.com/bit-hacks-part-3-playing-rightmost-set-bit-number/)
  4. [Bit Hacks — Part 4 (Playing with letters of English alphabet)](https://www.techiedelight.com/bit-hacks-part-4-playing-letters-english-alphabet/)
  5. [Bit Hacks — Part 5 (Find absolute value of an integer without branching)](https://www.techiedelight.com/bit-hacks-part-5-find-absolute-value-integer-without-branching/)
  6. [Bit Hacks — Part 6 (Random Problems)](https://www.techiedelight.com/bit-hacks-part-6-random-problems/)
  7. [Brian Kernighan’s Algorithm to count set bits in an integer](https://www.techiedelight.com/brian-kernighans-algorithm-count-set-bits-integer/)
  8. [Round up to the next highest power of 2](https://www.techiedelight.com/round-next-highest-power-2/)
  9. [Round up to the previous power of 2](https://www.techiedelight.com/round-previous-power-2/)
  10. [Compute parity of a number using lookup table](https://www.techiedelight.com/compute-parity-number-using-lookup-table/)
  11. [Count set bits using lookup table](https://www.techiedelight.com/count-set-bits-using-lookup-table/)
  12. [Find the minimum or maximum of two integers without using branching](https://www.techiedelight.com/find-minimum-maximum-two-integers-without-using-branching/)
  13. [Multiply 16-bit integers using 8-bit multiplier](https://www.techiedelight.com/multiply-16-bit-integers-using-8-bit-multiplier/)
  14. [Swap individual bits at given position in an integer](https://www.techiedelight.com/swap-individual-bits-given-position-integer/)
  15. [Check if given number is power of 4 or not](https://www.techiedelight.com/check-given-number-power-of-4/)
  16. [Check if given number is power of 8 or not](https://www.techiedelight.com/check-given-number-power-8-not/)
  17. [Reverse Bits of a given Integer](https://www.techiedelight.com/reverse-bits-of-given-integer/)
  18. [Find odd occurring element in an array in single traversal](https://www.techiedelight.com/find-odd-occurring-element-array-single-traversal/)
  19. [Find two odd occurring elements in an array without using any extra space](https://www.techiedelight.com/find-two-odd-occurring-element-array-without-extra-space/)
  20. [Swap two bits at given position in an integer](https://www.techiedelight.com/swap-two-bits-given-position-integer/)
  21. [Add binary representation of two integers](https://www.techiedelight.com/add-binary-representation-two-integers/)
  22. [Swap Adjacent Bits of a Number](https://www.techiedelight.com/swap-adjacent-bits-number/)
  23. [Print all distinct subsets of a given set](https://www.techiedelight.com/print-distinct-subsets-given-set/)
  24. [Perform Division of two numbers without using division operator (/)](https://www.techiedelight.com/perform-division-two-numbers-without-using-division-operator/)
  25. [Check if adjacent bits are set in binary representation of a given number](https://www.techiedelight.com/check-adjacent-bits-set-binary-representation-number/)
  26. [Conditionally negate a value without branching](https://www.techiedelight.com/conditionally-negate-value-without-branching/)
  27. [Find two duplicate elements in a limited range array (using XOR)](https://www.techiedelight.com/find-two-duplicate-elements-limited-range-array-using-xor/)
  28. [Reverse Bits of an integer using lookup table](https://www.techiedelight.com/reverse-bits-integer-using-lookup-table/)
  29. [Find missing number and duplicate elements in an array](https://www.techiedelight.com/find-missing-number-duplicate-elements-array/)
  30. [Circular shift on binary representation of an integer by k positions](https://www.techiedelight.com/circular-shift-integer-k-positions/)
  31. [Compute modulus division without division and modulo operator](https://www.techiedelight.com/compute-modulus-division-without-division-modulo-operator/)
  32. [Solve given set of problems without using multiplication or division operators](https://www.techiedelight.com/solve-problems-without-using-multiplication-division-operators/)
  33. [Find XOR of two numbers without using XOR operator](https://www.techiedelight.com/find-xor-two-numbers-without-using-xor-operator/)
  34. [Generate power set of a given set](https://www.techiedelight.com/generate-power-set-given-set/)
  35. [Huffman Coding](https://www.techiedelight.com/huffman-coding/)
  36. [Find missing number in array without using extra space](https://www.techiedelight.com/find-missing-number-array-without-extra-space/)
  37. [Find odd occurring element in logarithmic time](https://www.techiedelight.com/find-odd-occurring-element-logn-time/)
  38. [Find all odd occurring elements in an array having limited range of elements](https://www.techiedelight.com/find-odd-occurring-elements-array/)

# Binary Tree

* 1. [Check if two given binary trees are identical or not](https://www.techiedelight.com/check-if-two-binary-trees-are-identical-not-iterative-recursive/)
  2. [Calculate height of a binary tree](https://www.techiedelight.com/calculate-height-binary-tree-iterative-recursive/)
  3. [Delete given Binary Tree](https://www.techiedelight.com/delete-given-binary-tree-iterative-recursive/)
  4. [Inorder Tree Traversal (Iterative & Recursive Implementation)](https://www.techiedelight.com/inorder-tree-traversal-iterative-recursive/)
  5. [Preorder Tree Traversal (Iterative & Recursive Implementation)](https://www.techiedelight.com/preorder-tree-traversal-iterative-recursive/)
  6. [Postorder Tree Traversal (Iterative & Recursive Implementation)](https://www.techiedelight.com/postorder-tree-traversal-iterative-recursive/)
  7. [Level Order Traversal of Binary Tree](https://www.techiedelight.com/level-order-traversal-binary-tree/)
  8. [Spiral Order Traversal of Binary Tree](https://www.techiedelight.com/spiral-order-traversal-binary-tree/)
  9. [Reverse Level Order Traversal of Binary Tree](https://www.techiedelight.com/reverse-level-order-traversal-binary-tree/)
  10. [Print all nodes of a given binary tree in specific order](https://www.techiedelight.com/print-nodes-binary-tree-specific-order/)
  11. [Print left view of binary tree](https://www.techiedelight.com/print-left-view-of-binary-tree/)
  12. [Print Bottom View of Binary Tree](https://www.techiedelight.com/print-bottom-view-of-binary-tree/)
  13. [Print Top View of Binary Tree](https://www.techiedelight.com/print-top-view-binary-tree/)
  14. [Find next node in same level for given node in a binary tree](https://www.techiedelight.com/find-next-node-in-same-level-binary-tree/)
  15. [Check if given binary tree is complete binary tree or not](https://www.techiedelight.com/check-given-binary-tree-complete-binary-tree-not/)
  16. [In-place convert given binary tree to its sum tree](https://www.techiedelight.com/inplace-convert-a-tree-sum-tree/)
  17. [Determine if given two nodes are cousins of each other](https://www.techiedelight.com/determine-two-nodes-are-cousins/)
  18. [Print cousins of given node in a binary tree](https://www.techiedelight.com/print-cousins-of-given-node-binary-tree/)
  19. [Check if given binary tree is a sum tree or not](https://www.techiedelight.com/check-given-binary-tree-sum-tree-not/)
  20. [Combinations of words formed by replacing given numbers with corresponding alphabets](https://www.techiedelight.com/combinations-of-words-formed-replacing-given-numbers-corresponding-english-alphabet/)
  21. [Determine if given binary tree is a subtree of another binary tree or not](https://www.techiedelight.com/determine-given-binary-tree-is-subtree-of-another-binary-tree-not/)
  22. [Find diameter of a binary tree](https://www.techiedelight.com/find-diameter-of-a-binary-tree/)
  23. [Check if given binary Tree has symmetric structure or not](https://www.techiedelight.com/check-given-binary-tree-symmetric-structure-not/)
  24. [Convert binary tree to its mirror](https://www.techiedelight.com/convert-binary-tree-to-its-mirror/)
  25. [Check if binary tree can be converted to another by doing any no. of swaps of left & right child](https://www.techiedelight.com/determine-binary-tree-can-converted-another-number-swaps-left-right-child/)
  26. [Find Lowest Common Ancestor (LCA) of two nodes in a binary tree](https://www.techiedelight.com/find-lowest-common-ancestor-lca-two-nodes-binary-tree/)
  27. [Print all paths from root to leaf nodes in a binary tree](https://www.techiedelight.com/print-all-paths-from-root-to-leaf-nodes-binary-tree/)
  28. [Find ancestors of given node in a Binary Tree](https://www.techiedelight.com/find-ancestors-of-given-node-binary-tree/)
  29. [Find the distance between given pairs of nodes in a binary tree](https://www.techiedelight.com/distance-between-given-pairs-of-nodes-binary-tree/)
  30. [Find Vertical Sum in a given Binary Tree](https://www.techiedelight.com/find-vertical-sum-given-binary-tree/)
  31. [Perform vertical traversal of a binary tree — I](https://www.techiedelight.com/vertical-traversal-binary-tree/)
  32. [Perform vertical traversal of a binary tree — II](https://www.techiedelight.com/print-nodes-binary-tree-vertical-order/)
  33. [Print corner nodes of every level in binary tree](https://www.techiedelight.com/print-corner-nodes-every-level-binary-tree/)
  34. [Find the diagonal sum of given binary tree](https://www.techiedelight.com/find-diagonal-sum-given-binary-tree/)
  35. [Print Diagonal Traversal of Binary Tree](https://www.techiedelight.com/print-diagonal-traversal-binary-tree/)
  36. [In-place convert Binary Tree to Doubly Linked List](https://www.techiedelight.com/place-convert-given-binary-tree-to-doubly-linked-list/)
  37. [Sink nodes containing zero to the bottom of the binary tree](https://www.techiedelight.com/sink-nodes-containing-zero-bottom-binary-tree/)
  38. [Convert given binary tree to full tree by removing half nodes](https://www.techiedelight.com/convert-given-binary-tree-to-full-tree-removing-half-nodes/)
  39. [Truncate given binary tree to remove nodes which lie on a path having sum less than K](https://www.techiedelight.com/truncate-given-binary-tree-remove-nodes-lie-path-sum-less-k/)
  40. [Find maximum sum root-to-leaf path in a binary tree](https://www.techiedelight.com/find-maximum-sum-root-to-leaf-path-binary-tree/)
  41. [Check if given binary tree is height balanced or not](https://www.techiedelight.com/check-given-binary-tree-is-height-balanced-not/)
  42. [Find maximum width of given binary tree](https://www.techiedelight.com/find-maximum-width-given-binary-tree/)
  43. [Convert normal binary tree to Left-child right-sibling binary tree](https://www.techiedelight.com/convert-normal-binary-tree-left-child-right-sibling-binary-tree/)
  44. [Determine if given Binary Tree is a BST or not](https://www.techiedelight.com/determine-given-binary-tree-is-a-bst-or-not/)
  45. [Convert a Binary Tree to BST by maintaining its original structure](https://www.techiedelight.com/convert-binary-tree-to-bst-maintaining-original-structure/)
  46. [Invert a Binary Tree](https://www.techiedelight.com/invert-binary-tree-recursive-iterative/)
  47. [Print Right View of a Binary Tree](https://www.techiedelight.com/print-right-view-binary-tree/)
  48. [Print all paths from leaf to root node in given binary tree](https://www.techiedelight.com/print-all-paths-from-leaf-to-root-binary-tree/)
  49. [Iteratively print leaf to root path for every leaf node in a binary tree](https://www.techiedelight.com/print-leaf-to-root-path-binary-tree/)
  50. [Build Binary Tree from given Parent array](https://www.techiedelight.com/build-binary-tree-given-parent-array/)
  51. [Find all nodes at given distance from leaf nodes in a binary tree](https://www.techiedelight.com/find-all-nodes-at-given-distance-from-leaf-nodes-in-a-binary-tree/)
  52. [Count all subtrees having same value of nodes in a binary tree](https://www.techiedelight.com/count-subtrees-value-nodes-binary-tree/)
  53. [Find Maximum Difference Between a Node and its Descendants in a Binary Tree](https://www.techiedelight.com/find-maximum-difference-node-descendants/)
  54. [Construct a Binary Tree from Ancestor Matrix](https://www.techiedelight.com/construct-binary-tree-ancestor-matrix/)
  55. [Calculate height of a binary tree with leaf nodes forming a circular doubly linked list](https://www.techiedelight.com/calculate-height-binary-tree-leaf-nodes-forming-circular-doubly-linked-list/)
  56. [Find maximum sum path between two leaves in a binary tree](https://www.techiedelight.com/find-maximum-sum-path-between-two-leaves-in-a-binary-tree/)
  57. [Fix a binary tree that is only one swap away from becoming a BST](https://www.techiedelight.com/fix-binary-tree-one-swap-bst/)
  58. [Construct a binary tree from inorder and preorder traversal](https://www.techiedelight.com/construct-binary-tree-from-inorder-preorder-traversal/)
  59. [Construct a binary tree from inorder and postorder traversals](https://www.techiedelight.com/construct-binary-tree-from-inorder-postorder-traversals/)
  60. [Construct a binary tree from inorder and level order sequence](https://www.techiedelight.com/construct-binary-tree-from-inorder-level-order-traversals/)
  61. [Construct a full binary tree from preorder sequence with leaf node information](https://www.techiedelight.com/construct-full-binary-tree-from-preorder-sequence-with-leaf-information/)
  62. [Construct a full binary tree from a preorder and postorder sequence](https://www.techiedelight.com/construct-full-binary-tree-from-preorder-postorder-sequence/)
  63. [Set next pointer to inorder successor of all nodes in binary tree](https://www.techiedelight.com/set-next-pointer-inorder-successor-binary-tree/)
  64. [Efficiently print all nodes between two given levels in a binary tree](https://www.techiedelight.com/print-nodes-between-two-levels-binary-tree/)
  65. [Find preorder traversal of a binary tree from its inorder and postorder sequence](https://www.techiedelight.com/find-preorder-traversal-binary-tree-from-inorder-postorder/)
  66. [Find the difference between sum of all nodes present at odd and even levels in a binary tree](https://www.techiedelight.com/difference-between-sum-nodes-odd-even-levels/)
  67. [Find the size of the largest BST in a Binary Tree](https://www.techiedelight.com/find-size-largest-bst-in-binary-tree/)
  68. [Link nodes present in each level of a binary tree in the form of a linked list](https://www.techiedelight.com/link-nodes-each-level-binary-tree/)
  69. [Construct a Cartesian Tree from In-order Traversal](https://www.techiedelight.com/construct-cartesian-tree-from-inorder-traversal/)
  70. [Implementation of Treap Data Structure (Insert, Search and Delete)](https://www.techiedelight.com/implementation-treap-data-structure-cpp-java-insert-search-delete/)
  71. [Clone a binary tree with random pointers](https://www.techiedelight.com/clone-a-binary-tree-with-random-pointers/)
  72. [Threaded Binary Tree: Overview and Implementation](https://www.techiedelight.com/threaded-binary-tree-overview-implementation/)
  73. [Invert alternate levels of a perfect binary tree](https://www.techiedelight.com/invert-alternate-levels-perfect-binary-tree/)
  74. [Convert a Binary Tree into a Doubly Linked List in Spiral Order](https://www.techiedelight.com/convert-binary-tree-into-doubly-linked-list/)
  75. [Check if a binary tree is a min-heap or not](https://www.techiedelight.com/check-binary-tree-is-min-heap/)
  76. [Determine if a binary tree satisfy the height-balanced property of red–black tree](https://www.techiedelight.com/determine-binary-tree-satisfy-height-balanced-property-red-black-tree/)
  77. [Depth first search (DFS) vs Breadth first search (BFS)](https://www.techiedelight.com/depth-first-search-dfs-vs-breadth-first-search-bfs/)

# Matrix

* 1. [Print Matrix in Spiral Order](https://www.techiedelight.com/print-matrix-spiral-order/)
  2. [Create Spiral Matrix from given array](https://www.techiedelight.com/create-spiral-matrix-given-array/)
  3. [Shift all matrix elements by 1 in Spiral Order](https://www.techiedelight.com/shift-matrix-elements-1-spiral-order/)
  4. [Find Shortest path from source to destination in a matrix that satisfies given constraints](https://www.techiedelight.com/find-shortest-path-source-destination-matrix-satisfies-given-constraints/)
  5. [Change all elements of row i and column j in a matrix to 0 if cell (i, j) has value 0](https://www.techiedelight.com/change-elements-row-column-j-matrix-0-cell-j-value-0/)
  6. [Print diagonal elements of the matrix having positive slope](https://www.techiedelight.com/print-matrix-diagonally-positive-slope/)
  7. [Find all paths from first cell to last cell of a matrix](https://www.techiedelight.com/find-all-paths-from-source-to-destination-in-matrix/)
  8. [Replace all occurrences of 0 that are not surrounded by 1 in a binary matrix](https://www.techiedelight.com/replace-occurrences-0-not-surrounded-1-binary-matrix/)
  9. [In-place rotate the matrix by 90 degrees in clock-wise direction](https://www.techiedelight.com/place-rotate-matrix-90-degrees-clock-wise-direction/)
  10. [Count negative elements present in sorted matrix in linear time](https://www.techiedelight.com/count-negative-elements-present-sorted-matrix/)
  11. [Report all occurrences of an element in row wise and column wise sorted matrix in linear time](https://www.techiedelight.com/report-all-occurrences-of-an-element-in-sorted-matrix/)
  12. [Calculate sum of all elements in a sub-matrix in constant time](https://www.techiedelight.com/calculate-sum-elements-sub-matrix-constant-time/)
  13. [Find maximum sum K x K sub-matrix in a given M x N matrix](https://www.techiedelight.com/find-maximum-sum-submatrix-in-given-matrix/)
  14. [Find maximum sum submatrix present in a given matrix](https://www.techiedelight.com/find-maximum-sum-submatrix-present-given-matrix/)
  15. [Count the number of islands](https://www.techiedelight.com/count-the-number-of-islands/)
  16. [Flood Fill Algorithm](https://www.techiedelight.com/flood-fill-algorithm/)
  17. [Find shortest safe route in a field with sensors present](https://www.techiedelight.com/find-shortest-safe-route-field-sensors-present/)
  18. [Find all occurrences of given string in a character matrix](https://www.techiedelight.com/find-occurrences-given-string-character-matrix/)
  19. [Shortest path in a Maze | Lee Algorithm](https://www.techiedelight.com/lee-algorithm-shortest-path-in-a-maze/)
  20. [Check if given matrix is Toeplitz matrix or not](https://www.techiedelight.com/check-given-matrix-toeplitz-matrix-not/)
  21. [In-place rotate the matrix by 180 degrees](https://www.techiedelight.com/inplace-rotate-matrix-180-degrees/)
  22. [Fill Binary Matrix with Alternating Rectangles of 0 and 1](https://www.techiedelight.com/fill-binary-matrix-alternating-rectangles-0-1/)
  23. [Find all common elements present in every row of given matrix](https://www.techiedelight.com/find-common-elements-present-every-row-given-matrix/)
  24. [Construct a Binary Tree from Ancestor Matrix](https://www.techiedelight.com/construct-binary-tree-ancestor-matrix/)
  25. [Find common elements present in all rows of a matrix](https://www.techiedelight.com/find-common-elements-present-rows-matrix/)
  26. [Find index of the row containing maximum number of 1’s in a binary matrix](https://www.techiedelight.com/find-index-row-containing-maximum-number-1s-matrix/)
  27. [Find the largest square sub-matrix which is surrounded by all 1's](https://www.techiedelight.com/largest-square-sub-matrix-surrounded-by-1s/)
  28. [Find minimum passes required to convert all negative values in the matrix](https://www.techiedelight.com/find-minimum-passes-required-convert-negative-values-matrix/)
  29. [Print a spiral square matrix without using any extra space](https://www.techiedelight.com/print-spiral-square-matrix-without-extra-space/)
  30. [Print all shortest routes in a rectangular grid](https://www.techiedelight.com/print-all-shortest-routes-rectangular-grid/)
  31. [Find length of longest path in the matrix with consecutive characters](https://www.techiedelight.com/find-length-longest-path-matrix-consecutive-characters/)
  32. [Collect maximum value of coins in a matrix](https://www.techiedelight.com/collect-maximum-value-coins-matrix/)
  33. [Young Tableau | Insert, Search, Extract-Min, Delete, Replace](https://www.techiedelight.com/young-tableau-insert-search-extract-min-delete-replace/)
  34. [Sort an array using Young tableau](https://www.techiedelight.com/sort-array-using-young-tableau/)
  35. [Find path from source to destination in a matrix that satisfies given constraints](https://www.techiedelight.com/find-path-source-destination-matrix-satisfies-given-constraints/)
  36. [Generate list of possible words from a character matrix](https://www.techiedelight.com/generate-list-of-possible-words-from-a-character-matrix/)
  37. [Find probability that a person is alive after taking N steps on the island](https://www.techiedelight.com/probability-alive-after-taking-n-steps-island/)
  38. [Collect maximum points in a matrix by satisfying given constraints](https://www.techiedelight.com/collect-maximum-points-matrix-satisfying-given-constraints/)
  39. [Count number of paths in a matrix with given cost to reach destination cell](https://www.techiedelight.com/counting-paths-on-grid-to-reach-destination-cell/)
  40. [Find longest sequence formed by adjacent numbers in the matrix](https://www.techiedelight.com/find-longest-sequence-formed-adjacent-numbers-matrix/)
  41. [Find the minimum cost to reach last cell of the matrix from its first cell](https://www.techiedelight.com/find-minimum-cost-reach-last-cell-matrix-first-cell/)
  42. [Ways to reach the bottom-right corner of a matrix with exactly k turns allowed](https://www.techiedelight.com/reach-bottom-right-corner-matrix-exactly-k-turns/)
  43. [Matrix Chain Multiplication](https://www.techiedelight.com/matrix-chain-multiplication/)
  44. [Find size of largest square sub-matrix of 1’s present in given binary matrix](https://www.techiedelight.com/find-size-largest-square-sub-matrix-1s-present-given-binary-matrix/)
  45. [Chess Knight Problem — Find Shortest path from source to destination](https://www.techiedelight.com/chess-knight-problem-find-shortest-path-source-destination/)
  46. [Find Duplicate rows in a binary matrix](https://www.techiedelight.com/find-duplicate-rows-binary-matrix/)
  47. [Print all possible solutions to N Queens Problem](https://www.techiedelight.com/print-possible-solutions-n-queens-problem/)
  48. [Print all Possible Knight’s Tours in a chessboard](https://www.techiedelight.com/print-possible-knights-tours-chessboard/)
  49. [Find Shortest Path in Maze](https://www.techiedelight.com/find-shortest-path-in-maze/)
  50. [Find Longest Possible Route in a Matrix](https://www.techiedelight.com/find-longest-possible-route-matrix/)
  51. [Find total number of unique paths in a maze from source to destination](https://www.techiedelight.com/find-total-number-unique-paths-maze-source-destination/)
  52. [Calculate size of the largest plus of 1’s in binary matrix](https://www.techiedelight.com/calculate-size-largest-plus-1s-binary-matrix/)
  53. [Find the maximum value of M[c][d] — M[a][b] over all choices of indexes](https://www.techiedelight.com/find-maximum-value-choices-indexes/)
  54. [Find shortest distance of every cell from landmine in a Maze](https://www.techiedelight.com/find-shortest-distance-every-cell-landmine-maze/)
  55. [Find shortest route in a device to construct the given string](https://www.techiedelight.com/find-shortest-route-device-construct-given-string/)
  56. [Calculate minimum cost to reach destination city from source city](https://www.techiedelight.com/calculate-minimum-cost-to-reach-destination-city-from-source-city/)
  57. [Count all paths in a matrix from first cell to last cell](https://www.techiedelight.com/count-all-paths-matrix-from-first-cell-to-last-cell/)
  58. [Merge M sorted lists each containing N elements](https://www.techiedelight.com/merge-m-sorted-lists-containing-n-elements/)
  59. [Travelling Salesman Problem using Branch and Bound](https://www.techiedelight.com/travelling-salesman-problem-using-branch-and-bound/)

# Heap

* 1. [Introduction to Priority Queues using Binary Heaps](https://www.techiedelight.com/introduction-priority-queues-using-binary-heaps/)
  2. Min Heap and Max Heap Implementation — [C++](https://www.techiedelight.com/min-heap-max-heap-implementation-c/), [Java](https://www.techiedelight.com/min-heap-max-heap-implementation-in-java/)
  3. [Heap Sort Algorithm](https://www.techiedelight.com/heap-sort-place-place-implementation-c-c/)
  4. [Check if given array represents min heap or not](https://www.techiedelight.com/check-given-array-represents-min-heap-not/)
  5. [Convert Max Heap to Min Heap in linear time](https://www.techiedelight.com/convert-max-heap-min-heap-linear-time/)
  6. [Find K’th largest element in an array](https://www.techiedelight.com/find-kth-largest-element-array/)
  7. [Sort a K-Sorted Array](https://www.techiedelight.com/sort-k-sorted-array/)
  8. [Merge M sorted lists of variable length](https://www.techiedelight.com/merge-m-sorted-lists-variable-length/)
  9. [Merge K sorted linked lists](https://www.techiedelight.com/efficiently-merge-k-sorted-linked-lists/)
  10. [Find K’th smallest element in an array](https://www.techiedelight.com/find-kth-smallest-element-array/)
  11. [Find smallest range with at-least one element from each of the given lists](https://www.techiedelight.com/find-smallest-range-least-one-element-given-lists/)
  12. [Merge M sorted lists each containing N elements](https://www.techiedelight.com/merge-m-sorted-lists-containing-n-elements/)
  13. [Find first k non-repeating characters in a string in single traversal](https://www.techiedelight.com/first-k-non-repeating-characters-string/)
  14. [Find first k maximum occurring words in given set of strings](https://www.techiedelight.com/find-first-k-maximum-occurring-words-given-set-strings/)
  15. [Implementation of Treap Data Structure (Insert, Search and Delete)](https://www.techiedelight.com/implementation-treap-data-structure-cpp-java-insert-search-delete/)
  16. [Convert a Binary Search Tree into a Min Heap](https://www.techiedelight.com/convert-binary-search-tree-into-min-heap/)
  17. [Check if a binary tree is a min-heap or not](https://www.techiedelight.com/check-binary-tree-is-min-heap/)
  18. [Huffman Coding](https://www.techiedelight.com/huffman-coding/)
  19. [External Merge Sort Algorithm](https://www.techiedelight.com/external-merge-sort/)

# BST

* 1. [Insertion in BST](https://www.techiedelight.com/insertion-in-bst/)
  2. [Search given key in BST](https://www.techiedelight.com/search-given-key-in-bst/)
  3. [Deletion from BST](https://www.techiedelight.com/deletion-from-bst/)
  4. [Construct balanced BST from given keys](https://www.techiedelight.com/construct-balanced-bst-given-keys/)
  5. [Determine if given Binary Tree is a BST or not](https://www.techiedelight.com/determine-given-binary-tree-is-a-bst-or-not/)
  6. [Check if given keys represents same BSTs or not without building the BST](https://www.techiedelight.com/check-given-keys-represents-same-bsts-not-without-building-bst/)
  7. [Find inorder predecessor for given key in a BST](https://www.techiedelight.com/find-inorder-predecessor-given-key-bst/)
  8. [Find Lowest Common Ancestor (LCA) of two nodes in a Binary Search Tree](https://www.techiedelight.com/find-lowest-common-ancestor-lca-two-nodes-bst/)
  9. [Find K’th smallest and K’th largest element in BST](https://www.techiedelight.com/find-kth-smallest-largest-element-bst/)
  10. [Floor and Ceil in a Binary Search Tree](https://www.techiedelight.com/floor-ceil-bst-iterative-recursive/)
  11. [Find optimal cost to construct binary search tree](https://www.techiedelight.com/find-optimal-cost-to-construct-binary-search-tree/)
  12. [Convert a Binary Tree to BST by maintaining its original structure](https://www.techiedelight.com/convert-binary-tree-to-bst-maintaining-original-structure/)
  13. [Remove nodes from BST that have keys outside the valid range](https://www.techiedelight.com/remove-nodes-bst-keys-outside-valid-range/)
  14. [Find a pair with given sum in a BST](https://www.techiedelight.com/find-pair-with-given-sum-bst/)
  15. [Find inorder successor for given key in a BST](https://www.techiedelight.com/find-inorder-successor-given-key-bst/)
  16. [Replace every element of an array with the least greater element on its right](https://www.techiedelight.com/replace-every-element-array-least-greater-element-right/)
  17. [Fix a binary tree that is only one swap away from becoming a BST](https://www.techiedelight.com/fix-binary-tree-one-swap-bst/)
  18. [Update every key in BST to contain sum of all greater keys](https://www.techiedelight.com/update-every-key-bst-contain-sum-greater-keys/)
  19. [Check if a given sequence represents preorder traversal of a BST](https://www.techiedelight.com/check-sequence-represents-preorder-traversal-bst/)
  20. [Build a Binary Search Tree from a Postorder Sequence](https://www.techiedelight.com/build-binary-search-tree-from-postorder-sequence/)
  21. [Build a Binary Search Tree from a Preorder Sequence](https://www.techiedelight.com/build-binary-search-tree-from-preorder-sequence/)
  22. [Find a triplet with given sum in a BST](https://www.techiedelight.com/find-triplet-with-given-sum-bst/)
  23. [Count subtrees in a BST whose nodes lies within a given range](https://www.techiedelight.com/count-subtrees-bst-whose-nodes-within-range/)
  24. [Merge two BSTs into a doubly linked list in sorted order](https://www.techiedelight.com/merge-two-bsts-into-doubly-linked-list-sorted-order/)
  25. [Construct a height-balanced BST from an unbalanced BST](https://www.techiedelight.com/construct-height-balanced-bst-from-unbalanced-bst/)
  26. [Find the size of the largest BST in a Binary Tree](https://www.techiedelight.com/find-size-largest-bst-in-binary-tree/)
  27. [Convert a Binary Search Tree into a Min Heap](https://www.techiedelight.com/convert-binary-search-tree-into-min-heap/)
  28. [Construct a Height-Balanced BST from a Sorted Doubly Linked List](https://www.techiedelight.com/construct-height-balanced-bst-from-sorted-doubly-linked-list/)

# Trie

* 1. Trie Implementation — [C](https://www.techiedelight.com/trie-implementation-insert-search-delete/), [C++](https://www.techiedelight.com/cpp-implementation-trie-data-structure/), [Java](https://www.techiedelight.com/implement-trie-data-structure-java/), [Python](https://www.techiedelight.com/trie-implementation-python/)
  2. [Memory Efficient Implementation of Trie | Insert, Search and Delete](https://www.techiedelight.com/memory-efficient-trie-implementation-using-map-insert-search-delete/)
  3. [Longest Common Prefix in given set of strings (using Trie)](https://www.techiedelight.com/longest-common-prefix-given-set-strings-using-trie/)
  4. [Lexicographic sorting of given set of keys](https://www.techiedelight.com/lexicographic-sorting-given-set-of-keys/)
  5. [Find maximum occurring word in given set of strings](https://www.techiedelight.com/find-maximum-occurring-word-given-set-strings/)
  6. [Find first k maximum occurring words in given set of strings](https://www.techiedelight.com/find-first-k-maximum-occurring-words-given-set-strings/)
  7. [Find Duplicate rows in a binary matrix](https://www.techiedelight.com/find-duplicate-rows-binary-matrix/)
  8. [Word Break Problem | Using Trie](https://www.techiedelight.com/word-break-problem-using-trie/)
  9. [Generate list of possible words from a character matrix](https://www.techiedelight.com/generate-list-of-possible-words-from-a-character-matrix/)
  10. [Find all words matching a pattern in the given dictionary](https://www.techiedelight.com/find-all-words-matching-pattern-dictionary/)

# Sorting

* 1. [Insertion Sort Algorithm](https://www.techiedelight.com/insertion-sort-iterative-recursive/)
  2. [Selection Sort Algorithm](https://www.techiedelight.com/selection-sort-iterative-recursive/)
  3. [Bubble Sort Algorithm](https://www.techiedelight.com/bubble-sort-iterative-recursive/)
  4. [Merge Sort Algorithm](https://www.techiedelight.com/merge-sort/)
  5. [Iterative Merge Sort Algorithm (Bottom-up Merge Sort)](https://www.techiedelight.com/iterative-merge-sort-algorithm-bottom-up/)
  6. [QuickSort Algorithm](https://www.techiedelight.com/quicksort/)
  7. [Iterative Implementation of QuickSort](https://www.techiedelight.com/iterative-implementation-of-quicksort/)
  8. [Hybrid QuickSort](https://www.techiedelight.com/hybrid-quicksort/)
  9. [QuickSort using Dutch National Flag Algorithm](https://www.techiedelight.com/quicksort-using-dutch-national-flag-algorithm/)
  10. [QuickSort using Hoare’s Partitioning scheme](https://www.techiedelight.com/quick-sort-using-hoares-partitioning-scheme/)
  11. [Heap Sort Algorithm](https://www.techiedelight.com/heap-sort-place-place-implementation-c-c/)
  12. [Introsort Algorithm](https://www.techiedelight.com/introsort-algorithm/)
  13. [External Merge Sort Algorithm](https://www.techiedelight.com/external-merge-sort/)
  14. [Counting Sort Algorithm](https://www.techiedelight.com/counting-sort-algorithm-implementation/)
  15. [Inversion Count of an array](https://www.techiedelight.com/inversion-count-array/)
  16. [Sort an array using Young tableau](https://www.techiedelight.com/sort-array-using-young-tableau/)
  17. [Merge Sort Algorithm for Singly Linked List](https://www.techiedelight.com/merge-sort-singly-linked-list/)
  18. [Problems solved using partitioning logic of QuickSort](https://www.techiedelight.com/problems-solved-using-partitioning-logic-quicksort/)
  19. [Sort a Doubly Linked List using Merge Sort](https://www.techiedelight.com/sort-doubly-linked-list-merge-sort/)
  20. [Sort elements by their frequency and Index](https://www.techiedelight.com/sort-elements-by-their-frequency-and-index/)
  21. [Sort an array based on order defined by another array](https://www.techiedelight.com/custom-sort-sort-elements-array-order-elements-defined-second-array/)
  22. [Efficiently sort an array with many duplicated values](https://www.techiedelight.com/efficiently-sort-array-duplicated-values/)
  23. [Find largest number possible from set of given numbers](https://www.techiedelight.com/find-largest-number-possible-set-given-numbers/)
  24. [Find the surpasser count for each element of an array](https://www.techiedelight.com/surpasser-count-each-element-array/)
  25. [Segregate positive and negative integers using Merge Sort](https://www.techiedelight.com/segregate-positive-negative-integers-using-mergesort/)
  26. [Group anagrams together from given list of words](https://www.techiedelight.com/group-anagrams-together-given-list-words/)

# Divide & Conquer

* 1. [Binary Search Algorithm](https://www.techiedelight.com/binary-search/)
  2. [Find number of rotations in a circularly sorted array](https://www.techiedelight.com/find-number-rotations-circularly-sorted-array/)
  3. [Search an element in a circular sorted array](https://www.techiedelight.com/search-element-circular-sorted-array/)
  4. [Find first or last occurrence of a given number in a sorted array](https://www.techiedelight.com/find-first-or-last-occurrence-of-a-given-number-sorted-array/)
  5. [Count occurrences of a number in a sorted array with duplicates](https://www.techiedelight.com/count-occurrences-number-sorted-array-duplicates/)
  6. [Find smallest missing element from a sorted array](https://www.techiedelight.com/find-smallest-missing-element-sorted-array/)
  7. [Find Floor and Ceil of a number in a sorted array](https://www.techiedelight.com/find-floor-ceil-number-sorted-array/)
  8. [Search in a nearly sorted array in logarithmic time](https://www.techiedelight.com/search-nearly-sorted-array-ologn-time/)
  9. [Find number of 1’s in a sorted binary array](https://www.techiedelight.com/find-number-1s-sorted-binary-array/)
  10. [Find the peak element in an array](https://www.techiedelight.com/find-peak-element-array/)
  11. [Maximum Sum Subarray using Divide & Conquer](https://www.techiedelight.com/maximum-sum-subarray-using-divide-conquer/)
  12. [Efficiently implement a power function](https://www.techiedelight.com/power-function-implementation-recursive-iterative/)
  13. [Find Missing Term in a Sequence in Logarithmic time](https://www.techiedelight.com/find-missing-term-sequence-ologn-time/)
  14. [Division of Two Numbers using Binary Search Algorithm](https://www.techiedelight.com/division-two-numbers-using-binary-search-algorithm/)
  15. [Find Floor and Ceil of a number in a sorted array (Recursive solution)](https://www.techiedelight.com/find-floor-ceil-number-sorted-array-recursive/)
  16. [Find Frequency of each element in a sorted array containing duplicates](https://www.techiedelight.com/find-frequency-element-sorted-array-containing-duplicates/)
  17. [Find odd occurring element in logarithmic time](https://www.techiedelight.com/find-odd-occurring-element-logn-time/)
  18. [Ternary Search vs Binary search](https://www.techiedelight.com/ternary-search-vs-binary-search/)
  19. [Exponential search](https://www.techiedelight.com/exponential-search/)
  20. [Unbounded Binary Search](https://www.techiedelight.com/unbounded-binary-search/)
  21. [Interpolation search](https://www.techiedelight.com/interpolation-search/)
  22. [Merge Sort Algorithm](https://www.techiedelight.com/merge-sort/)
  23. [QuickSort Algorithm](https://www.techiedelight.com/quicksort/)

# Greedy

* 1. [Activity Selection Problem](https://www.techiedelight.com/activity-selection-problem/)
  2. [Huffman Coding](https://www.techiedelight.com/huffman-coding/)
  3. [Job Sequencing Problem with Deadlines](https://www.techiedelight.com/job-sequencing-problem-deadlines/)
  4. [Graph Coloring Problem](https://www.techiedelight.com/greedy-coloring-graph/)
  5. [Kruskal’s Algorithm for finding Minimum Spanning Tree](https://www.techiedelight.com/kruskals-algorithm-for-finding-minimum-spanning-tree/)
  6. [Single-Source Shortest Paths — Dijkstra’s Algorithm](https://www.techiedelight.com/single-source-shortest-paths-dijkstras-algorithm/)
  7. [Shortest Superstring Problem](https://www.techiedelight.com/shortest-superstring-problem/)

# Dynamic Programming

* 1. [Introduction to Dynamic Programming](https://www.techiedelight.com/introduction-dynamic-programming/)
  2. [Longest Common Subsequence Problem](https://www.techiedelight.com/longest-common-subsequence/)
  3. [Longest Common Subsequence | Space optimized version](https://www.techiedelight.com/longest-common-subsequence-lcs-space-optimized-version/)
  4. [Longest Common Subsequence of K-sequences](https://www.techiedelight.com/longest-common-subsequence-of-k-sequences/)
  5. [Longest Common Subsequence | Finding all LCS](https://www.techiedelight.com/longest-common-subsequence-finding-lcs/)
  6. [Longest Common Substring Problem](https://www.techiedelight.com/longest-common-substring-problem/)
  7. [Longest Palindromic Subsequence Problem](https://www.techiedelight.com/longest-palindromic-subsequence-using-dynamic-programming/)
  8. [Longest Repeated Subsequence Problem](https://www.techiedelight.com/longest-repeated-subsequence-problem/)
  9. [Implement Diff Utility](https://www.techiedelight.com/implement-diff-utility/)
  10. [Shortest Common Supersequence Problem](https://www.techiedelight.com/shortest-common-supersequence-introduction-scs-length/)
  11. [Shortest Common Supersequence | Finding all SCS](https://www.techiedelight.com/shortest-common-supersequence-finding-scs/)
  12. [Shortest Common Supersequence Problem using LCS](https://www.techiedelight.com/shortest-common-supersequence-using-lcs/)
  13. [Longest Increasing Subsequence Problem](https://www.techiedelight.com/longest-increasing-subsequence-using-dynamic-programming/)
  14. [Longest Decreasing Subsequence Problem](https://www.techiedelight.com/longest-decreasing-subsequence-problem/)
  15. [Longest Bitonic Subsequence](https://www.techiedelight.com/longest-bitonic-subsequence/)
  16. [Increasing Subsequence with Maximum Sum](https://www.techiedelight.com/increasing-subsequence-with-maximum-sum/)
  17. [The Levenshtein Distance (Edit Distance) Problem](https://www.techiedelight.com/levenshtein-distance-edit-distance-problem/)
  18. [Find size of largest square sub-matrix of 1’s present in given binary matrix](https://www.techiedelight.com/find-size-largest-square-sub-matrix-1s-present-given-binary-matrix/)
  19. [Matrix Chain Multiplication](https://www.techiedelight.com/matrix-chain-multiplication/)
  20. [Find the minimum cost to reach last cell of the matrix from its first cell](https://www.techiedelight.com/find-minimum-cost-reach-last-cell-matrix-first-cell/)
  21. [Find longest sequence formed by adjacent numbers in the matrix](https://www.techiedelight.com/find-longest-sequence-formed-adjacent-numbers-matrix/)
  22. [Count number of paths in a matrix with given cost to reach destination cell](https://www.techiedelight.com/counting-paths-on-grid-to-reach-destination-cell/)
  23. [0–1 Knapsack Problem](https://www.techiedelight.com/0-1-knapsack-problem/)
  24. [Maximize value of the expression](https://www.techiedelight.com/maximize-value-of-the-expression/)
  25. [Partition Problem](https://www.techiedelight.com/partition-problem/)
  26. [Subset sum Problem](https://www.techiedelight.com/subset-sum-problem/)
  27. [3-Partition Problem](https://www.techiedelight.com/3-partition-problem/)
  28. [Minimum Sum Partition Problem](https://www.techiedelight.com/minimum-sum-partition-problem/)
  29. [Rod Cutting](https://www.techiedelight.com/rot-cutting/)
  30. [Maximum Product Rod Cutting](https://www.techiedelight.com/maximum-product-rod-cutting/)
  31. [Coin change-making problem (unlimited supply of coins)](https://www.techiedelight.com/coin-change-making-problem-unlimited-supply-coins/)
  32. [Coin Change Problem — Find total number of ways to get the denomination of coins](https://www.techiedelight.com/coin-change-problem-find-total-number-ways-get-denomination-coins/)
  33. [Total possible solutions to linear equation of k variables](https://www.techiedelight.com/total-possible-solutions-linear-equation-k-variables/)
  34. [Longest Alternating Subsequence Problem](https://www.techiedelight.com/longest-alternating-subsequence/)
  35. [Count number of times a pattern appears in given string as a subsequence](https://www.techiedelight.com/count-number-times-pattern-appears-given-string-subsequence/)
  36. [Collect maximum points in a matrix by satisfying given constraints](https://www.techiedelight.com/collect-maximum-points-matrix-satisfying-given-constraints/)
  37. [Find all N-digit binary strings without any consecutive 1’s](https://www.techiedelight.com/find-n-digit-binary-strings-without-consecutive-1s/)
  38. [Count total possible combinations of N-digit numbers in a mobile keypad](https://www.techiedelight.com/count-total-possible-combinations-n-digit-numbers-mobile-keypad/)
  39. [Word Break Problem](https://www.techiedelight.com/word-break-problem/)
  40. [Determine Minimal Adjustment Cost of an Array](https://www.techiedelight.com/determine-minimal-adjustment-cost-array/)
  41. [Check if a string is K-Palindrome or not](https://www.techiedelight.com/check-given-string-k-palindrome-not/)
  42. [Find total ways to achieve given sum with n throws of dice having k faces](https://www.techiedelight.com/total-ways-sum-with-n-throws-dice-having-k-faces/)
  43. [Wildcard Pattern Matching](https://www.techiedelight.com/wildcard-pattern-matching/)
  44. [Find number of ways to fill a N x 4 matrix with 1 x 4 tiles](https://www.techiedelight.com/find-ways-fill-n-x-4-matrix-with-1-x-4-tiles/)
  45. [Ways to reach the bottom-right corner of a matrix with exactly k turns allowed](https://www.techiedelight.com/reach-bottom-right-corner-matrix-exactly-k-turns/)
  46. [Weighted Interval Scheduling Problem](https://www.techiedelight.com/weighted-interval-scheduling-problem/)
  47. [Box Stacking Problem](https://www.techiedelight.com/box-stacking-problem/)
  48. [Find total ways to reach the n’th stair with at-most m steps](https://www.techiedelight.com/find-total-ways-reach-nth-stair-with-atmost-m-steps/)
  49. [Find total ways to reach the n’th stair from the bottom](https://www.techiedelight.com/find-total-ways-to-reach-nth-stair/)
  50. [Activity Selection Problem](https://www.techiedelight.com/activity-selection-problem-using-dynamic-programming/)
  51. [Find minimum number of deletions required to convert a string into palindrome](https://www.techiedelight.com/find-minimum-number-deletions-convert-string-into-palindrome/)
  52. [Calculate minimum cost to reach destination city from source city](https://www.techiedelight.com/calculate-minimum-cost-to-reach-destination-city-from-source-city/)
  53. [Pots of Gold Game Problem](https://www.techiedelight.com/pots-gold-game-dynamic-programming/)
  54. [Find minimum cuts needed for palindromic partition of a string](https://www.techiedelight.com/find-minimum-cuts-needed-palindromic-partition-string/)
  55. [Weighted Interval Scheduling using LIS algorithm](https://www.techiedelight.com/weighted-interval-scheduling-problem-using-lis/)
  56. [Find minimum jumps required to reach the destination](https://www.techiedelight.com/find-minimum-jumps-required-reach-destination/)
  57. [Find probability that a person is alive after taking N steps on the island](https://www.techiedelight.com/probability-alive-after-taking-n-steps-island/)
  58. [Find maximum sum of subsequence with no adjacent elements](https://www.techiedelight.com/maximum-sum-of-subsequence-with-no-adjacent-elements/)
  59. [Maximum Length Snake Sequence](https://www.techiedelight.com/maximum-length-snake-sequence/)
  60. [Calculate size of the largest plus of 1’s in binary matrix](https://www.techiedelight.com/calculate-size-largest-plus-1s-binary-matrix/)
  61. [Longest Increasing Subsequence using LCS](https://www.techiedelight.com/longest-increasing-subsequence-using-lcs/)
  62. [Find maximum profit earned from at most K stock transactions](https://www.techiedelight.com/find-maximum-profit-earned-at-most-k-stock-transactions/)
  63. [Count all paths in a matrix from first cell to last cell](https://www.techiedelight.com/count-all-paths-matrix-from-first-cell-to-last-cell/)
  64. [Check if a string matches with a given wildcard pattern](https://www.techiedelight.com/check-string-matches-with-wildcard-pattern/)
  65. [Check if given string is interleaving of two other given strings](https://www.techiedelight.com/check-string-interleaving-two-given-strings/)
  66. [Find all employees who directly or indirectly reports to a manager](https://www.techiedelight.com/find-employees-who-reports-to-manager/)
  67. [Find optimal cost to construct binary search tree](https://www.techiedelight.com/find-optimal-cost-to-construct-binary-search-tree/)
  68. [Find maximum sum of subsequence with no adjacent elements](https://www.techiedelight.com/maximum-sum-of-subsequence-with-no-adjacent-elements/)
  69. [Maximum Sum Subarray Problem (Kadane’s Algorithm)](https://www.techiedelight.com/maximum-subarray-problem-kadanes-algorithm/)
  70. [Longest Alternating Subarray Problem](https://www.techiedelight.com/longest-alternating-subarray-problem/)
  71. [Collect maximum value of coins in a matrix](https://www.techiedelight.com/collect-maximum-value-coins-matrix/)
  72. [Find length of longest path in the matrix with consecutive characters](https://www.techiedelight.com/find-length-longest-path-matrix-consecutive-characters/)
  73. [Find ways to calculate a target from elements of specified array](https://www.techiedelight.com/find-ways-calculate-target-elements-array/)
  74. [Calculate sum of all elements in a sub-matrix in constant time](https://www.techiedelight.com/calculate-sum-elements-sub-matrix-constant-time/)
  75. [Find maximum sum K x K sub-matrix in a given M x N matrix](https://www.techiedelight.com/find-maximum-sum-submatrix-in-given-matrix/)
  76. [Find maximum sum submatrix present in a given matrix](https://www.techiedelight.com/find-maximum-sum-submatrix-present-given-matrix/)
  77. [Single-Source Shortest Paths — Bellman Ford Algorithm](https://www.techiedelight.com/single-source-shortest-paths-bellman-ford-algorithm/)
  78. [All-Pairs Shortest Paths — Floyd Warshall Algorithm](https://www.techiedelight.com/pairs-shortest-paths-floyd-warshall-algorithm/)

# Backtracking

* 1. [Print all possible solutions to N Queens Problem](https://www.techiedelight.com/print-possible-solutions-n-queens-problem/)
  2. [Print all Possible Knight’s Tours in a chessboard](https://www.techiedelight.com/print-possible-knights-tours-chessboard/)
  3. [Find Shortest Path in Maze](https://www.techiedelight.com/find-shortest-path-in-maze/)
  4. [Find Longest Possible Route in a Matrix](https://www.techiedelight.com/find-longest-possible-route-matrix/)
  5. [Find path from source to destination in a matrix that satisfies given constraints](https://www.techiedelight.com/find-path-source-destination-matrix-satisfies-given-constraints/)
  6. [Find total number of unique paths in a maze from source to destination](https://www.techiedelight.com/find-total-number-unique-paths-maze-source-destination/)
  7. [Print All Hamiltonian Path present in a graph](https://www.techiedelight.com/print-all-hamiltonian-path-present-in-a-graph/)
  8. [Print all k-colorable configurations of the graph (Vertex coloring of graph)](https://www.techiedelight.com/print-k-colorable-configurations-graph-vertex-coloring-graph/)
  9. [Find all Permutations of a given string](https://www.techiedelight.com/find-permutations-given-string/)
  10. [All combinations of elements satisfying given constraints](https://www.techiedelight.com/find-combinations-of-elements-satisfies-given-constraints/)
  11. [Find all binary strings that can be formed from given wildcard pattern](https://www.techiedelight.com/find-binary-strings-can-formed-given-wildcard-pattern/)
  12. [K-Partition Problem | Printing all Partitions](https://www.techiedelight.com/k-partition-problem-print-all-subsets/)
  13. [Magnet Puzzle](https://www.techiedelight.com/magnet-puzzle/)
  14. [Find ways to calculate a target from elements of specified array](https://www.techiedelight.com/find-ways-calculate-target-elements-array/)
  15. [Find minimum number possible by doing at-most K swaps](https://www.techiedelight.com/find-minimum-number-possible-k-swaps/)
  16. [Determine if a pattern matches with a string or not](https://www.techiedelight.com/determine-pattern-matches-string-not/)
  17. [Generate list of possible words from a character matrix](https://www.techiedelight.com/generate-list-of-possible-words-from-a-character-matrix/)
  18. [Find the path between given vertices in a directed graph](https://www.techiedelight.com/find-path-between-vertices-directed-graph/)
  19. [Find all Possible Topological Orderings of a DAG](https://www.techiedelight.com/find-all-possible-topological-orderings-of-dag/)
  20. [Print all shortest routes in a rectangular grid](https://www.techiedelight.com/print-all-shortest-routes-rectangular-grid/)

# Graph

* 1. [Terminology and Representations of Graphs](https://www.techiedelight.com/terminology-and-representations-of-graphs/)
  2. Graph Implementation — [C](https://www.techiedelight.com/implement-graph-data-structure-c/), [C++](https://www.techiedelight.com/graph-implementation-c-without-using-stl/), [C++ (STL)](https://www.techiedelight.com/graph-implementation-using-stl/), [Java (Collections)](https://www.techiedelight.com/graph-implementation-java-using-collections/), [Python](https://www.techiedelight.com/graph-implementation-python/)
  3. [Breadth First Search (BFS) Algorithm](https://www.techiedelight.com/breadth-first-search/)
  4. [Depth First Search (DFS) Algorithm](https://www.techiedelight.com/depth-first-search/)
  5. [Depth first search (DFS) vs Breadth first search (BFS)](https://www.techiedelight.com/depth-first-search-dfs-vs-breadth-first-search-bfs/)
  6. [Arrival and Departure Time of Vertices in DFS](https://www.techiedelight.com/arrival-departure-time-vertices-dfs/)
  7. [Types of edges involved in DFS and relation between them](https://www.techiedelight.com/types-edges-involved-dfs-relation/)
  8. [Bipartite Graph](https://www.techiedelight.com/bipartite-graph/)
  9. [Determine if a given graph is Bipartite Graph using DFS](https://www.techiedelight.com/determine-given-graph-bipartite-graph-using-dfs/)
  10. [Snake and Ladder Problem](https://www.techiedelight.com/min-throws-required-to-win-snake-and-ladder-game/)
  11. [Topological Sorting in a DAG](https://www.techiedelight.com/topological-sorting-dag/)
  12. [Kahn’s Topological Sort Algorithm](https://www.techiedelight.com/kahn-topological-sort-algorithm/)
  13. [Transitive Closure of a Graph](https://www.techiedelight.com/transitive-closure-graph/)
  14. [Check if an undirected graph contains cycle or not](https://www.techiedelight.com/check-undirected-graph-contains-cycle-not/)
  15. [Total paths in given digraph from given source to destination having exactly m edges](https://www.techiedelight.com/total-paths-in-digraph-from-source-to-destination-m-edges/)
  16. [Determine if an undirected graph is a Tree (Acyclic Connected Graph)](https://www.techiedelight.com/determine-undirected-graph-tree-acyclic-connected-graph/)
  17. [2-Edge Connectivity in the graph](https://www.techiedelight.com/2-edge-connectivity-graph/)
  18. [2-Vertex Connectivity in the graph](https://www.techiedelight.com/2-vertex-connectivity-graph/)
  19. [Check if given digraph is a DAG (Directed Acyclic Graph) or not](https://www.techiedelight.com/check-given-digraph-dag-directed-acyclic-graph-not/)
  20. [Disjoint-Set Data Structure (Union-Find Algorithm)](https://www.techiedelight.com/disjoint-set-data-structure-union-find-algorithm/)
  21. [Chess Knight Problem — Find Shortest path from source to destination](https://www.techiedelight.com/chess-knight-problem-find-shortest-path-source-destination/)
  22. [Check if given Graph is Strongly Connected or not](https://www.techiedelight.com/check-given-graph-strongly-connected-not/)
  23. [Check if given Graph is Strongly Connected or not using one DFS Traversal](https://www.techiedelight.com/check-graph-strongly-connected-one-dfs-traversal/)
  24. [Union-Find Algorithm for Cycle Detection in undirected graph](https://www.techiedelight.com/union-find-algorithm-cycle-detection-graph/)
  25. [Kruskal’s Algorithm for finding Minimum Spanning Tree](https://www.techiedelight.com/kruskals-algorithm-for-finding-minimum-spanning-tree/)
  26. [Single-Source Shortest Paths — Dijkstra’s Algorithm](https://www.techiedelight.com/single-source-shortest-paths-dijkstras-algorithm/)
  27. [Single-Source Shortest Paths — Bellman Ford Algorithm](https://www.techiedelight.com/single-source-shortest-paths-bellman-ford-algorithm/)
  28. [All-Pairs Shortest Paths — Floyd Warshall Algorithm](https://www.techiedelight.com/pairs-shortest-paths-floyd-warshall-algorithm/)
  29. [Find Cost of Shortest Path in DAG using one pass of Bellman-Ford](https://www.techiedelight.com/cost-of-shortest-path-in-dag-using-one-pass-of-bellman-ford/)
  30. [Least Cost Path in Weighted Digraph using BFS](https://www.techiedelight.com/least-cost-path-weighted-digraph-using-bfs/)
  31. [Find maximum cost path in graph from given source to destination](https://www.techiedelight.com/maximum-cost-path-graph-source-destination/)
  32. [Determine negative-weight cycle in a graph](https://www.techiedelight.com/determine-negative-weight-cycle-graph/)
  33. [Least cost path in given digraph from given source to destination having exactly m edges](https://www.techiedelight.com/least-cost-path-digraph-source-destination-m-edges/)
  34. [Find the path between given vertices in a directed graph](https://www.techiedelight.com/find-path-between-vertices-directed-graph/)
  35. [Find all Possible Topological Orderings of a DAG](https://www.techiedelight.com/find-all-possible-topological-orderings-of-dag/)
  36. [Find the correct order of alphabets in a given dictionary of ancient origin](https://www.techiedelight.com/find-correct-order-alphabets-dictionary-ancient-origin/)
  37. [Find longest path in a Directed Acyclic Graph (DAG)](https://www.techiedelight.com/find-cost-longest-path-dag/)
  38. [Construct a directed graph from undirected graph that satisfies given constraints](https://www.techiedelight.com/construct-directed-graph-from-undirected-graph/)
  39. [Print all k-colorable configurations of the graph (Vertex coloring of graph)](https://www.techiedelight.com/print-k-colorable-configurations-graph-vertex-coloring-graph/)
  40. [Print All Hamiltonian Path present in a graph](https://www.techiedelight.com/print-all-hamiltonian-path-present-in-a-graph/)
  41. [Graph Coloring Problem](https://www.techiedelight.com/greedy-coloring-graph/)

# Puzzles

* 1. [Clock Angle Problem — Find angle between hour and minute hand](https://www.techiedelight.com/angle-between-hour-minute-hand/)
  2. [Add two numbers without using addition operator](https://www.techiedelight.com/add-two-numbers-without-using-addition-operator/)
  3. [Generate power set of a given set](https://www.techiedelight.com/generate-power-set-given-set/)
  4. [Implement power function without using multiplication and division operators](https://www.techiedelight.com/implement-power-function-without-using-multiplication-division-operators/)
  5. [Print all numbers between 1 to N without using semicolon](https://www.techiedelight.com/print-numbers-1-n-without-using-semicolon/)
  6. [Swap two numbers without using third variable](https://www.techiedelight.com/swap-two-numbers-without-using-third-variable/)
  7. [Determine the if condition to print specific output](https://www.techiedelight.com/determine-condition-to-print-specific-output/)
  8. [Find maximum & minimum of triplet without using conditional statement and ternary operator](https://www.techiedelight.com/maximum-minimum-three-numbers-without-using-conditional-statement-ternary-operator/)
  9. [Find numbers represented as sum of two cubes for two different pairs](https://www.techiedelight.com/numbers-represented-as-sum-of-two-cubes/)
  10. [Print “Hello World” with empty main() function](https://www.techiedelight.com/print-hello-world-empty-main-function/)
  11. [Tower of Hanoi Problem](https://www.techiedelight.com/tower-of-hanoi-problem/)
  12. [Print all numbers between 1 to N without using any loop](https://www.techiedelight.com/print-numbers-1-n-without-using-loop-4-methods/)
  13. [Print a semicolon without using semicolon anywhere in the program](https://www.techiedelight.com/print-a-semicolon-without-using-semicolon-anywhere-program/)
  14. [Multiply two numbers without using multiplication operator or loops](https://www.techiedelight.com/multiply-two-numbers-without-using-multiplication-operator-loops/)
  15. [Find square of a number without using multiplication and division operator](https://www.techiedelight.com/find-square-number-without-using-multiplication-division-operator/)
  16. [Find if a number is even or odd without using any conditional statement](https://www.techiedelight.com/find-number-even-odd-without-using-conditional-statement/)
  17. [Set both elements of a binary array to 0 in single line](https://www.techiedelight.com/set-elements-binary-array-0-single-line/)
  18. [Find minimum number without using conditional statement or ternary operator](https://www.techiedelight.com/find-minimum-number-without-using-conditional-statement-ternary-operator/)
  19. [Perform Division of two numbers without using division operator (/)](https://www.techiedelight.com/perform-division-two-numbers-without-using-division-operator/)
  20. [Generate 0 and 1 with 75% and 25% Probability](https://www.techiedelight.com/generate-0-1-75-25-probability/)
  21. [Generate Desired Random Numbers With Equal Probability](https://www.techiedelight.com/generate-random-numbers-equal-probability/)
  22. [Return 0, 1 and 2 with equal Probability using the specified function](https://www.techiedelight.com/return-0-1-2-equal-probability-using-specified-function/)
  23. [Generate Fair Results from a Biased Coin](https://www.techiedelight.com/generate-fair-results-biased-coin/)
  24. [Generate numbers from 1 to 7 with equal probability using specified function](https://www.techiedelight.com/generate-numbers-1-7-equal-probability/)
  25. [Implement Ternary Operator Without Using Conditional Expressions](https://www.techiedelight.com/implement-ternary-operator-without-using-conditional-statements/)
  26. [Determine if two integers are equal without using comparison and arithmetic operators](https://www.techiedelight.com/determine-two-integers-equal-without-using-comparison-arithmetic-operators/)
  27. [Return 0 and 1 with equal Probability using the specified function](https://www.techiedelight.com/get-0-1-equal-probability-using-specified-function/)
  28. [Generate random input from an array according to given probabilities](https://www.techiedelight.com/generate-random-input-array-according-given-probabilities/)
  29. [Compute modulus division without division and modulo operator](https://www.techiedelight.com/compute-modulus-division-without-division-modulo-operator/)

# Source: <https://medium.com/techie-delight/500-data-structures-and-algorithms-practice-problems-35afe8a1e222>