SINGLE LINKED LISTS

1. Introduction to Linked List,
2. Linked List vs Array,
3. Linked List Insertion,
4. Linked List Deletion,
5. A Programmer’s approach of looking at Array vs. Linked List,
6. Find Length of a Linked List (Iterative and Recursive),
7. How to write C functions that modify head pointer of a Linked List?,
8. Write a function to get Nth node in a Linked List,
9. Given only a pointer to a node to be deleted in a singly linked list, how do you delete it?,
10. Print the middle of a given linked list,
11. Nth node from the end of a Linked List,
12. Write a function to delete a Linked List,
13. Write a function that counts the number of times a given int occurs in a Linked List,
14. Reverse a linked list,
15. Detect loop in a linked list,
16. Function to check if a singly linked list is palindrome,
17. Given a linked list which is sorted, how will you insert in sorted way,
18. Intersection point of two Linked Lists.,
19. Recursive function to print reverse of a Linked List,
20. Remove duplicates from a sorted linked list,
21. Remove duplicates from an unsorted linked list,
22. Pairwise swap elements of a given linked list,
23. Practice questions for Linked List and Recursion,
24. Move last element to front of a given Linked List,
25. Intersection of two Sorted Linked Lists,
26. Delete alternate nodes of a Linked List,
27. Alternating split of a given Singly Linked List,
28. Merge two sorted linked lists,
29. Identical Linked List
30. Merge Sort for Linked Lists,
31. Reverse a Linked List in groups of given size,
32. Reverse alternate K nodes in a Singly Linked List,
33. Delete nodes which have a greater value on right side,
34. Segregate even and odd nodes in a Linked List,
35. Detect and Remove Loop in a Linked List,
36. Add two numbers represented by linked lists | Set 1,
37. Delete a given node in Linked List under given constraints,
38. Union and Intersection of two Linked Lists,
39. Find a triplet from three linked lists with sum equal to a given number,
40. Rotate a Linked List,
41. Flattening a Linked List,
42. Add two numbers represented by linked lists | Set 2,
43. Sort a linked list of 0s, 1s and 2s,
44. Flatten a multilevel linked list,
45. Delete N nodes after M nodes of a linked list,
46. QuickSort on Singly Linked List,
47. Merge a linked list into another linked list at alternate positions,
48. Pairwise swap elements of a given linked list by changing links,
49. Given a linked list of line segments, remove middle points,
50. Construct a Maximum Sum Linked List out of two Sorted Linked Lists having some Common nodes,
51. Can we reverse a linked list in less than O(n)?,
52. Clone a linked list with next and random pointer | Set 2

**Circular Linked List*:***

1. [Circular Linked List Introduction and Applications,](http://geeksquiz.com/circular-linked-list/)
2. [Circular Linked List Traversal](http://geeksquiz.com/circular-linked-list-set-2-traversal/)
3. [Split a Circular Linked List into two halves](http://www.geeksforgeeks.org/split-a-circular-linked-list-into-two-halves/)
4. [Sorted insert for circular linked list](http://www.geeksforgeeks.org/sorted-insert-for-circular-linked-list/)

**Doubly Linked List:**

1. [Doubly Linked List Introduction and Insertion](http://geeksquiz.com/doubly-linked-list/),
2. [Delete a node in a Doubly Linked List](http://www.geeksforgeeks.org/delete-a-node-in-a-doubly-linked-list/),
3. [Reverse a Doubly Linked List](http://www.geeksforgeeks.org/reverse-a-doubly-linked-list/),
4. [The Great Tree-List Recursion Problem.](http://www.geeksforgeeks.org/the-great-tree-list-recursion-problem/),
5. [Copy a linked list with next and arbit pointer](http://www.geeksforgeeks.org/a-linked-list-with-next-and-arbit-pointer/),
6. [QuickSort on Doubly Linked List](http://www.geeksforgeeks.org/quicksort-for-linked-list/),
7. [Swap Kth node from beginning with Kth node from end in a Linked List](http://www.geeksforgeeks.org/swap-kth-node-from-beginning-with-kth-node-from-end-in-a-linked-list/)

**Stack:**

1. [Introduction to Stack](http://geeksquiz.com/stack-set-1/)
2. [Infix to Postfix Conversion using Stack](http://geeksquiz.com/stack-set-2-infix-to-postfix/),
3. [Evaluation of Postfix Expression](http://geeksquiz.com/stack-set-4-evaluation-postfix-expression/),
4. [Reverse a Sting using Stack](http://geeksquiz.com/stack-set-3-reverse-string-using-stack/),
5. [Implement two stacks in an array](http://www.geeksforgeeks.org/archives/18754),
6. [Check for balanced parentheses in an expression](http://www.geeksforgeeks.org/archives/6547),
7. [Next Greater Element](http://www.geeksforgeeks.org/archives/8405),
8. [Reverse a stack using recursion](http://www.geeksforgeeks.org/archives/6921),
9. [The Stock Span Problem](http://www.geeksforgeeks.org/the-stock-span-problem/),
10. [Design and Implement Special Stack Data Structure](http://www.geeksforgeeks.org/design-and-implement-special-stack-data-structure/),
11. [Implement Stack using Queues](http://www.geeksforgeeks.org/implement-stack-using-queue/),
12. [Design a stack with operations on middle element](http://www.geeksforgeeks.org/design-a-stack-with-find-middle-operation/),
13. [How to create mergable stack?](http://geeksquiz.com/create-mergable-stack/),
14. [How to efficiently implement k stacks in a single array?](http://www.geeksforgeeks.org/efficiently-implement-k-stacks-single-array/)

**Binary Tree:**

1. [Binary Tree Introduction](http://geeksquiz.com/binary-tree-set-1-introduction/),
2. [Binary Tree Properties](http://geeksquiz.com/binary-tree-set-2-properties/),
3. [Types of Binary Tree](http://geeksquiz.com/binary-tree-set-3-types-of-binary-tree/),
4. [Applications of tree data structure](http://www.geeksforgeeks.org/applications-of-tree-data-structure/),
5. [Tree Traversals](http://www.geeksforgeeks.org/618/),
6. [Threaded Binary Tree](http://geeksquiz.com/threaded-binary-tree/),
7. [Size of a tree](http://www.geeksforgeeks.org/write-a-c-program-to-calculate-size-of-a-tree/),
8. [Determine if Two Trees are Identical](http://www.geeksforgeeks.org/write-c-code-to-determine-if-two-trees-are-identical/),
9. [Maximum Depth or Height of a Tree](http://www.geeksforgeeks.org/write-a-c-program-to-find-the-maximum-depth-or-height-of-a-tree/),
10. [Write a C program to Delete a Tree.](http://www.geeksforgeeks.org/write-a-c-program-to-delete-a-tree/),
11. [Write an Efficient C Function to Convert a Binary Tree into its Mirror Tree](http://www.geeksforgeeks.org/write-an-efficient-c-function-to-convert-a-tree-into-its-mirror-tree/),
12. [If you are given two traversal sequences, can you construct the binary tree?](http://www.geeksforgeeks.org/if-you-are-given-two-traversal-sequences-can-you-construct-the-binary-tree/),
13. [Given a binary tree, print out all of its root-to-leaf paths one per line.](http://www.geeksforgeeks.org/given-a-binary-tree-print-out-all-of-its-root-to-leaf-paths-one-per-line/),
14. [The Great Tree-List Recursion Problem.](http://www.geeksforgeeks.org/the-great-tree-list-recursion-problem/),
15. [Level Order Tree Traversal](http://www.geeksforgeeks.org/level-order-tree-traversal/),
16. [Count leaf nodes in a binary tree](http://www.geeksforgeeks.org/write-a-c-program-to-get-count-of-leaf-nodes-in-a-binary-tree/),
17. [Level order traversal in spiral form](http://www.geeksforgeeks.org/level-order-traversal-in-spiral-form/),
18. [Check for Children Sum Property in a Binary Tree.](http://www.geeksforgeeks.org/check-for-children-sum-property-in-a-binary-tree/),
19. [Convert an arbitrary Binary Tree to a tree that holds Children Sum Property](http://www.geeksforgeeks.org/convert-an-arbitrary-binary-tree-to-a-tree-that-holds-children-sum-property/),
20. [Diameter of a Binary Tree](http://www.geeksforgeeks.org/diameter-of-a-binary-tree/),
21. [How to determine if a binary tree is height-balanced?](http://www.geeksforgeeks.org/how-to-determine-if-a-binary-tree-is-balanced/),
22. [Inorder Tree Traversal without Recursion](http://www.geeksforgeeks.org/inorder-tree-traversal-without-recursion/),
23. [Inorder Tree Traversal without recursion and without stack!](http://www.geeksforgeeks.org/inorder-tree-traversal-without-recursion-and-without-stack/),
24. [Root to leaf path sum equal to a given number](http://www.geeksforgeeks.org/root-to-leaf-path-sum-equal-to-a-given-number/),
25. [Construct Tree from given Inorder and Preorder traversals](http://www.geeksforgeeks.org/construct-tree-from-given-inorder-and-preorder-traversal/),
26. [Given a binary tree, print all root-to-leaf paths](http://www.geeksforgeeks.org/given-a-binary-tree-print-all-root-to-leaf-paths/),
27. [Double Tree](http://www.geeksforgeeks.org/double-tree/),
28. [Maximum width of a binary tree](http://www.geeksforgeeks.org/maximum-width-of-a-binary-tree/),
29. [Foldable Binary Trees](http://www.geeksforgeeks.org/foldable-binary-trees/),
30. [Print nodes at k distance from root](http://www.geeksforgeeks.org/print-nodes-at-k-distance-from-root/),
31. [Get Level of a node in a Binary Tree](http://www.geeksforgeeks.org/get-level-of-a-node-in-a-binary-tree/),
32. [Print Ancestors of a given node in Binary Tree](http://www.geeksforgeeks.org/print-ancestors-of-a-given-node-in-binary-tree/),
33. [Check if a given Binary Tree is SumTree](http://www.geeksforgeeks.org/check-if-a-given-binary-tree-is-sumtree/),
34. [Check if a binary tree is subtree of another binary tree](http://www.geeksforgeeks.org/check-if-a-binary-tree-is-subtree-of-another-binary-tree/),
35. [Connect nodes at same level](http://www.geeksforgeeks.org/connect-nodes-at-same-level/),
36. [Connect nodes at same level using constant extra space](http://www.geeksforgeeks.org/connect-nodes-at-same-level-with-o1-extra-space/),
37. [Populate Inorder Successor for all nodes](http://www.geeksforgeeks.org/populate-inorder-successor-for-all-nodes/),
38. [Convert a given tree to its Sum Tree](http://www.geeksforgeeks.org/convert-a-given-tree-to-sum-tree/),
39. [Vertical Sum in a given Binary Tree](http://www.geeksforgeeks.org/vertical-sum-in-a-given-binary-tree/),
40. [Find the maximum sum leaf to root path in a Binary Tree](http://www.geeksforgeeks.org/find-the-maximum-sum-path-in-a-binary-tree/),
41. [Construct Special Binary Tree from given Inorder traversal](http://www.geeksforgeeks.org/construct-binary-tree-from-inorder-traversal/),
42. [Construct a special tree from given preorder traversal](http://www.geeksforgeeks.org/construct-a-special-tree-from-given-preorder-traversal/),
43. [Check whether a given Binary Tree is Complete or not](http://www.geeksforgeeks.org/check-if-a-given-binary-tree-is-complete-tree-or-not/),
44. [Boundary Traversal of binary tree](http://www.geeksforgeeks.org/boundary-traversal-of-binary-tree/),
45. [Construct Full Binary Tree from given preorder and postorder traversals](http://www.geeksforgeeks.org/full-and-complete-binary-tree-from-given-preorder-and-postorder-traversals/),
46. [Iterative Preorder Traversal](http://www.geeksforgeeks.org/iterative-preorder-traversal/),
47. [Morris traversal for Preorder](http://www.geeksforgeeks.org/morris-traversal-for-preorder/),
48. [Linked complete binary tree & its creation](http://www.geeksforgeeks.org/linked-complete-binary-tree-its-creation/),
49. [Ternary Search Tree](http://www.geeksforgeeks.org/ternary-search-tree/),
50. [Segment Tree | Set 1 (Sum of given range)](http://www.geeksforgeeks.org/segment-tree-set-1-sum-of-given-range/),
51. [Largest Independent Set Problem](http://www.geeksforgeeks.org/largest-independent-set-problem/),
52. [Iterative Postorder Traversal | Set 1 (Using Two Stacks)](http://www.geeksforgeeks.org/iterative-postorder-traversal/),
53. [Iterative Postorder Traversal | Set 2 (Using One Stack)](http://www.geeksforgeeks.org/iterative-postorder-traversal-using-stack/),
54. [Reverse Level Order Traversal](http://www.geeksforgeeks.org/reverse-level-order-traversal/),
55. [Construct Complete Binary Tree from its Linked List Representation](http://www.geeksforgeeks.org/given-linked-list-representation-of-complete-tree-convert-it-to-linked-representation/),
56. [Convert a given Binary Tree to Doubly Linked List | Set 1](http://www.geeksforgeeks.org/in-place-convert-a-given-binary-tree-to-doubly-linked-list/),
57. [Tree Isomorphism Problem](http://www.geeksforgeeks.org/tree-isomorphism-problem/),
58. [Find all possible interpretations of an array of digits](http://www.geeksforgeeks.org/find-all-possible-interpretations/),
59. [Iterative Method to find Height of Binary Tree](http://www.geeksforgeeks.org/iterative-method-to-find-height-of-binary-tree/),
60. [Custom Tree Problem](http://www.geeksforgeeks.org/custom-tree-problem/),
61. [Convert a given Binary Tree to Doubly Linked List | Set 2](http://www.geeksforgeeks.org/convert-a-given-binary-tree-to-doubly-linked-list-set-2/),
62. [Print ancestors of a given binary tree node without recursion](http://www.geeksforgeeks.org/print-ancestors-of-a-given-binary-tree-node-without-recursion/),
63. [Difference between sums of odd level and even level nodes of a Binary Tree](http://www.geeksforgeeks.org/difference-between-sums-of-odd-and-even-levels/),
64. [Print Postorder traversal from given Inorder and Preorder traversals](http://www.geeksforgeeks.org/print-postorder-from-given-inorder-and-preorder-traversals/),
65. [Find depth of the deepest odd level leaf node](http://www.geeksforgeeks.org/find-depth-of-the-deepest-odd-level-node/),
66. [Check if all leaves are at same level](http://www.geeksforgeeks.org/check-leaves-level/),
67. [Print Left View of a Binary Tree](http://www.geeksforgeeks.org/print-left-view-binary-tree/),
68. [Remove all nodes which don’t lie in any path with sum>= k](http://www.geeksforgeeks.org/remove-all-nodes-which-lie-on-a-path-having-sum-less-than-k/),
69. [Extract Leaves of a Binary Tree in a Doubly Linked List](http://www.geeksforgeeks.org/connect-leaves-doubly-linked-list/),
70. [Deepest left leaf node in a binary tree](http://www.geeksforgeeks.org/deepest-left-leaf-node-in-a-binary-tree/),
71. [Find next right node of a given key](http://www.geeksforgeeks.org/find-next-right-node-of-a-given-key/),
72. [Sum of all the numbers that are formed from root to leaf paths](http://www.geeksforgeeks.org/sum-numbers-formed-root-leaf-paths/),
73. [Convert a given Binary Tree to Doubly Linked List | Set 3](http://www.geeksforgeeks.org/convert-given-binary-tree-doubly-linked-list-set-3/),
74. [Lowest Common Ancestor in a Binary Tree | Set 1](http://www.geeksforgeeks.org/lowest-common-ancestor-binary-tree-set-1/),
75. [Find distance between two given keys of a Binary Tree](http://www.geeksforgeeks.org/find-distance-two-given-nodes/),
76. [Print all nodes that are at distance k from a leaf node](http://www.geeksforgeeks.org/print-nodes-distance-k-leaf-node/),
77. [Check if a given Binary Tree is height balanced like a Red-Black Tree,](http://www.geeksforgeeks.org/print-nodes-distance-k-given-node-binary-tree/)
78. [Print all nodes at distance k from a given node](http://www.geeksforgeeks.org/print-nodes-distance-k-given-node-binary-tree/),
79. [Print a Binary Tree in Vertical Order | Set 1](http://www.geeksforgeeks.org/print-nodes-distance-k-given-node-binary-tree/" \o "Permanent link to Print all nodes at distance k from a given node)
80. [Print all nodes at distance k from a given node](http://www.geeksforgeeks.org/print-nodes-distance-k-given-node-binary-tree/" \o "Permanent link to Print all nodes at distance k from a given node)
81. [Construct a tree from Inorder and Level order traversals](http://www.geeksforgeeks.org/construct-tree-inorder-level-order-traversals/),
82. [Find the maximum path sum between two leaves of a binary tree](http://www.geeksforgeeks.org/find-maximum-path-sum-two-leaves-binary-tree/),
83. [Reverse alternate levels of a perfect binary tree](http://www.geeksforgeeks.org/reverse-alternate-levels-binary-tree/),
84. [Check if two nodes are cousins in a Binary Tree](http://www.geeksforgeeks.org/check-two-nodes-cousins-binary-tree/),
85. [Check if a binary tree is subtree of another binary tree | Set 2](http://www.geeksforgeeks.org/check-binary-tree-subtree-another-binary-tree-set-2/),
86. [Serialize and Deserialize a Binary Tree](http://www.geeksforgeeks.org/serialize-deserialize-binary-tree/),
87. [Print nodes between two given level numbers of a binary tree](http://www.geeksforgeeks.org/given-binary-tree-print-nodes-two-given-level-numbers/),
88. [closest leaf in a Binary Tree](http://www.geeksforgeeks.org/find-closest-leaf-binary-tree/),
89. [Convert a Binary Tree to Threaded binary tree](http://www.geeksforgeeks.org/convert-binary-tree-threaded-binary-tree/),
90. [Print Nodes in Top View of Binary Tree](http://www.geeksforgeeks.org/print-nodes-top-view-binary-tree/),
91. [Bottom View of a Binary Tree](http://www.geeksforgeeks.org/bottom-view-binary-tree/),
92. [Perfect Binary Tree Specific Level Order Traversal](http://www.geeksforgeeks.org/perfect-binary-tree-specific-level-order-traversal/),
93. [Convert left-right representation of a bianry tree to down-right](http://geeksquiz.com/convert-left-right-representation-bianry-tree-right/),
94. [Print level order traversal line by line](http://geeksquiz.com/print-level-order-traversal-line-line/),
95. [Minimum no. of iterations to pass information to all nodes in the tree](http://www.geeksforgeeks.org/minimum-iterations-pass-information-nodes-tree/),
96. [Clone a Binary Tree with Random Pointers](http://www.geeksforgeeks.org/clone-binary-tree-random-pointers/),
97. [Given a binary tree, how do you remove all the half nodes?](http://www.geeksforgeeks.org/given-a-binary-tree-how-do-you-remove-all-the-half-nodes/),
98. [Vertex Cover Problem | Set 2 (Dynamic Programming Solution for Tree)](http://www.geeksforgeeks.org/vertex-cover-problem-set-2-dynamic-programming-solution-tree/),
99. [Check whether a binary tree is a full binary tree or not](http://www.geeksforgeeks.org/check-whether-binary-tree-full-binary-tree-not/),
100. [Find sum of all left leaves in a given Binary Tree](http://www.geeksforgeeks.org/find-sum-left-leaves-given-binary-tree/),
101. [Remove nodes on root to leaf paths of length < K](http://www.geeksforgeeks.org/remove-nodes-root-leaf-paths-length-k/)

**Binary Search Tree:**

1. [Search and Insert in BST](http://geeksquiz.com/binary-search-tree-set-1-search-and-insertion/),
2. [Deletion from BST](http://geeksquiz.com/binary-search-tree-set-2-delete/),
3. [Minimum value in a Binary Search Tree](http://www.geeksforgeeks.org/find-the-minimum-element-in-a-binary-search-tree/),
4. [Inorder predecessor and successor for a given key in BST](http://www.geeksforgeeks.org/inorder-predecessor-successor-given-key-bst/" \o "Permanent link to Inorder predecessor and successor for a given key in BST),
5. [Check if a binary tree is BST or not](http://www.geeksforgeeks.org/a-program-to-check-if-a-binary-tree-is-bst-or-not/),
6. [Lowest Common Ancestor in a Binary Search Tree.](http://www.geeksforgeeks.org/lowest-common-ancestor-in-a-binary-search-tree/)
7. [Sorted order printing of a given array that represents a BST](http://www.geeksforgeeks.org/sorted-order-printing-of-an-array-that-represents-a-bst/),
8. [Inorder Successor in Binary Search Tree](http://www.geeksforgeeks.org/inorder-successor-in-binary-search-tree/),
9. [Find k-th smallest element in BST (Order Statistics in BST)](http://www.geeksforgeeks.org/find-k-th-smallest-element-in-bst-order-statistics-in-bst/),
10. [Print BST keys in the given range](http://www.geeksforgeeks.org/print-bst-keys-in-the-given-range/),
11. [Sorted Array to Balanced BST](http://www.geeksforgeeks.org/sorted-array-to-balanced-bst/),
12. [Find the largest BST subtree in a given Binary Tree](http://www.geeksforgeeks.org/find-the-largest-subtree-in-a-tree-that-is-also-a-bst/),
13. [Check for Identical BSTs without building the trees](http://www.geeksforgeeks.org/check-for-identical-bsts-without-building-the-trees/),
14. [Add all greater values to every node in a given BST](http://www.geeksforgeeks.org/add-greater-values-every-node-given-bst/),
15. [Remove BST keys outside the given range](http://www.geeksforgeeks.org/remove-bst-keys-outside-the-given-range/),
16. [Check if each internal node of a BST has exactly one child](http://www.geeksforgeeks.org/check-if-each-internal-node-of-a-bst-has-exactly-one-child/),
17. [Find if there is a triplet in a Balanced BST that adds to zero](http://www.geeksforgeeks.org/find-if-there-is-a-triplet-in-bst-that-adds-to-0/),
18. [Merge two BSTs with limited extra space](http://www.geeksforgeeks.org/merge-two-bsts-with-limited-extra-space/),
19. [Two nodes of a BST are swapped, correct the BST](http://www.geeksforgeeks.org/fix-two-swapped-nodes-of-bst/),
20. [Construct BST from given preorder traversal | Set 1](http://www.geeksforgeeks.org/construct-bst-from-given-preorder-traversa/),
21. [Construct BST from given preorder traversal | Set 2](http://www.geeksforgeeks.org/construct-bst-from-given-preorder-traversal-set-2/),
22. [Floor and Ceil from a BST](http://www.geeksforgeeks.org/floor-and-ceil-from-a-bst/),
23. [Convert a BST to a Binary Tree such that sum of all greater keys is added to every key](http://www.geeksforgeeks.org/convert-bst-to-a-binary-tree/),
24. [Sorted Linked List to Balanced BST](http://www.geeksforgeeks.org/sorted-linked-list-to-balanced-bst/),
25. [In-place conversion of Sorted DLL to Balanced BST](http://www.geeksforgeeks.org/in-place-conversion-of-sorted-dll-to-balanced-bst/),
26. [Find a pair with given sum in a Balanced BST](http://www.geeksforgeeks.org/find-a-pair-with-given-sum-in-bst/),
27. [Total number of possible Binary Search Trees with n keys](http://www.geeksforgeeks.org/g-fact-18/),
28. [Merge Two Balanced Binary Search Trees](http://www.geeksforgeeks.org/merge-two-balanced-binary-search-trees/),
29. [Binary Tree to Binary Search Tree Conversion](http://www.geeksforgeeks.org/binary-tree-to-binary-search-tree-conversion/),
30. [Transform a BST to greater sum tree](http://www.geeksforgeeks.org/transform-bst-sum-tree/),
31. [Inorder predecessor and successor for a given key in BST](http://www.geeksforgeeks.org/inorder-predecessor-successor-given-key-bst/" \o "Permanent link to Inorder predecessor and successor for a given key in BST),
32. [K’th Largest Element in BST when modification to BST is not allowed](http://www.geeksforgeeks.org/kth-largest-element-in-bst-when-modification-to-bst-is-not-allowed/)
33. [Quiz on Binary Search Trees](http://geeksquiz.com/data-structure/binary-search-trees/)
34. [Quiz on Balanced Binary Search Trees](http://geeksquiz.com/data-structure/balanced-binary-search-trees/)

**Heap:**

1. [Binary Heap](http://geeksquiz.com/binary-heap/),
2. [Binomial Heap](http://www.geeksforgeeks.org/binomial-heap-2/),
3. [Heap Sort](http://geeksquiz.com/heap-sort/),
4. [K’th Largest Element in an array](http://www.geeksforgeeks.org/k-largestor-smallest-elements-in-an-array/),
5. [Sort an almost sorted array/](http://www.geeksforgeeks.org/nearly-sorted-algorithm/),
6. [Sort an almost sorted array/](http://www.geeksforgeeks.org/nearly-sorted-algorithm/),
7. [Tournament Tree (Winner Tree) and Binary Heap](http://www.geeksforgeeks.org/tournament-tree-and-binary-heap/)

**Hashing:**

1. [Hashing Introduction](http://geeksquiz.com/hashing-set-1-introduction/),
2. [Print a Binary Tree in Vertical Order](http://www.geeksforgeeks.org/print-binary-tree-vertical-order-set-2/),
3. [Find whether an array is subset of another array](http://www.geeksforgeeks.org/find-whether-an-array-is-subset-of-another-array-set-1/),
4. [Union and Intersection of two Linked Lists](http://www.geeksforgeeks.org/union-and-intersection-of-two-linked-lists/),
5. [Find a pair with given sum](http://www.geeksforgeeks.org/write-a-c-program-that-given-a-set-a-of-n-numbers-and-another-number-x-determines-whether-or-not-there-exist-two-elements-in-s-whose-sum-is-exactly-x/).
6. [Check if a given array contains duplicate elements within k distance from each other](http://www.geeksforgeeks.org/check-given-array-contains-duplicate-elements-within-k-distance/),
7. [Quiz on Hashing](http://geeksquiz.com/data-structure/hash/).

**Graph:**

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

* [Graph and its representations](http://www.geeksforgeeks.org/graph-and-its-representations/),
* [Breadth First Traversal for a Graph](http://www.geeksforgeeks.org/breadth-first-traversal-for-a-graph/),
* [Depth First Traversal for a Graph](http://www.geeksforgeeks.org/depth-first-traversal-for-a-graph/),
* [Applications of Depth First Search](http://www.geeksforgeeks.org/applications-of-depth-first-search/),
* [Applications of Breadth First Traversal](http://www.geeksforgeeks.org/applications-of-breadth-first-traversal/) ,
* [Detect Cycle in a Directed Graph](http://www.geeksforgeeks.org/detect-cycle-in-a-graph/),
* [Detect Cycle in a an Undirected Graph](http://www.geeksforgeeks.org/union-find/),
* [Detect cycle in an undirected graph](http://www.geeksforgeeks.org/detect-cycle-undirected-graph/),
* [Longest Path in a Directed Acyclic Graph](http://www.geeksforgeeks.org/find-longest-path-directed-acyclic-graph/),[Topological Sorting](http://www.geeksforgeeks.org/topological-sorting/),
* [Check whether a given graph is Bipartite or not](http://www.geeksforgeeks.org/bipartite-graph/),
* [Snake and Ladder Problem](http://www.geeksforgeeks.org/snake-ladder-problem-2/),
* [Minimize Cash Flow among a given set of friends who have borrowed money from each other](http://www.geeksforgeeks.org/minimize-cash-flow-among-given-set-friends-borrowed-money/),
* [Boggle (Find all possible words in a board of characters)](http://www.geeksforgeeks.org/boggle-find-possible-words-board-characters/)

**Minimum Spanning Tree:**

* [Prim’s Minimum Spanning Tree (MST))](http://www.geeksforgeeks.org/greedy-algorithms-set-5-prims-minimum-spanning-tree-mst-2/),
* [Applications of Minimum Spanning Tree Problem](http://www.geeksforgeeks.org/applications-of-minimum-spanning-tree/),
* [Prim’s MST for Adjacency List Representation](http://www.geeksforgeeks.org/greedy-algorithms-set-5-prims-mst-for-adjacency-list-representation/),
* [Kruskal’s Minimum Spanning Tree Algorithm](http://www.geeksforgeeks.org/greedy-algorithms-set-2-kruskals-minimum-spanning-tree-mst/)

***Shortest Paths:***

* [Dijkstra’s shortest path algorithm](http://www.geeksforgeeks.org/greedy-algorithms-set-6-dijkstras-shortest-path-algorithm/),
* [Dijkstra’s Algorithm for Adjacency List Representation](http://www.geeksforgeeks.org/greedy-algorithms-set-7-dijkstras-algorithm-for-adjacency-list-representation/),

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

***Connectivity:***

1. [Find if there is a path between two vertices in a directed graph](http://www.geeksforgeeks.org/find-if-there-is-a-path-between-two-vertices-in-a-given-graph/),
2. [Connectivity in a directed graph](http://www.geeksforgeeks.org/connectivity-in-a-directed-graph/),
3. [Articulation Points (or Cut Vertices) in a Graph](http://www.geeksforgeeks.org/articulation-points-or-cut-vertices-in-a-graph/),
4. [Biconnected graph](http://www.geeksforgeeks.org/biconnectivity-in-a-graph/), [Bridges in a graph](http://www.geeksforgeeks.org/bridge-in-a-graph/),
5. [Eulerian path and circuit](http://www.geeksforgeeks.org/eulerian-path-and-circuit/),
6. [Fleury’s Algorithm for printing Eulerian Path or Circuit](http://www.geeksforgeeks.org/fleurys-algorithm-for-printing-eulerian-path/),
7. [Strongly Connected Components](http://www.geeksforgeeks.org/strongly-connected-components/),
8. [Transitive closure of a graph](http://www.geeksforgeeks.org/transitive-closure-of-a-graph/),
9. [Find the number of islands](http://www.geeksforgeeks.org/find-number-of-islands/),
10. [Count all possible walks from a source to a destination with exactly k edges](http://www.geeksforgeeks.org/count-possible-paths-source-destination-exactly-k-edges/),
11. [Euler Circuit in a Directed Graph](http://www.geeksforgeeks.org/euler-circuit-directed-graph/),
12. [Biconnected Components](http://www.geeksforgeeks.org/biconnected-components/),
13. [Check if a given graph is tree or not](http://geeksquiz.com/check-given-graph-tree/)

***Hard Problems:***

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

***Maximum Flow:***

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

[Quiz on Graph](http://geeksquiz.com/data-structure/graph/)

[Quiz on Graph Traversals](http://geeksquiz.com/algorithms/graph-traversals/)

[Quiz on Graph Shortest Paths](http://geeksquiz.com/algorithms/graph-shortest-paths/)

[Quiz on Graph Minimum Spanning Tree](http://geeksquiz.com/algorithms/graph-minimum-spanning-tree/)

**Advanced Data Structure:**

***Advanced Lists:***

1. [Memory efficient doubly linked list](http://www.geeksforgeeks.org/memory-efficient-doubly-linked-list/),
2. [XOR Linked List – A Memory Efficient Doubly Linked List | Set 1](http://www.geeksforgeeks.org/xor-linked-list-a-memory-efficient-doubly-linked-list-set-1/),
3. [XOR Linked List – A Memory Efficient Doubly Linked List | Set 2](http://www.geeksforgeeks.org/xor-linked-list-a-memory-efficient-doubly-linked-list-set-2/),
4. [Skip List | Set 1 (Introduction)](http://www.geeksforgeeks.org/skip-list/),
5. [Self Organizing List | Set 1 (Introduction)](http://www.geeksforgeeks.org/self-organizing-list-set-1-introduction/)

***Trie:***

1. [Trie | (Insert and Search)](http://www.geeksforgeeks.org/trie-insert-and-search/),
2. [Trie | (Delete)](http://www.geeksforgeeks.org/trie-delete/),
3. [Longest prefix matching – A Trie based solution in Java](http://www.geeksforgeeks.org/longest-prefix-matching-a-trie-based-solution-in-java/),
4. [Print unique rows in a given boolean matrix](http://www.geeksforgeeks.org/print-unique-rows/).
5. [How to Implement Reverse DNS Look Up Cache?](http://www.geeksforgeeks.org/implement-reverse-dns-look-cache/),
6. [How to Implement Forward DNS Look Up Cache?](http://www.geeksforgeeks.org/implement-forward-dns-look-cache/),

***Suffix Array and Suffix Tree***:

1. [Suffix Array Introduction](http://www.geeksforgeeks.org/suffix-array-set-1-introduction/),
2. [Suffix Array nLogn Algorithm](http://www.geeksforgeeks.org/suffix-array-set-2-a-nlognlogn-algorithm/),
3. [Suffix Tree Introduction](http://www.geeksforgeeks.org/pattern-searching-set-8-suffix-tree-introduction/),
4. [Ukkonen’s Suffix Tree Construction – Part 1](http://www.geeksforgeeks.org/ukkonens-suffix-tree-construction-part-1/),
5. [Ukkonen’s Suffix Tree Construction – Part 2](http://www.geeksforgeeks.org/ukkonens-suffix-tree-construction-part-2/" \o "Permanent link to Ukkonen’s Suffix Tree Construction – Part 2),
6. [Ukkonen’s Suffix Tree Construction – Part 3](http://www.geeksforgeeks.org/ukkonens-suffix-tree-construction-part-3/),
7. [Ukkonen’s Suffix Tree Construction – Part 4,](http://www.geeksforgeeks.org/ukkonens-suffix-tree-construction-part-4/)
8. [Ukkonen’s Suffix Tree Construction – Part 5](http://www.geeksforgeeks.org/ukkonens-suffix-tree-construction-part-5/),
9. [Ukkonen’s Suffix Tree Construction – Part 6](http://www.geeksforgeeks.org/ukkonens-suffix-tree-construction-part-6/),
10. [Generalized Suffix Tree](http://www.geeksforgeeks.org/generalized-suffix-tree-1/),
11. [Build Linear Time Suffix Array using Suffix Tree](http://www.geeksforgeeks.org/suffix-tree-application-4-build-linear-time-suffix-array/),[Substring Check](http://www.geeksforgeeks.org/suffix-tree-application-1-substring-check/),
12. [Searching All Patterns](http://www.geeksforgeeks.org/suffix-tree-application-2-searching-all-patterns/),
13. [Longest Repeated Substring,](http://www.geeksforgeeks.org/suffix-tree-application-3-longest-repeated-substring/)
14. [Longest Common Substring, Longest Palindromic Substring](http://www.geeksforgeeks.org/suffix-tree-application-6-longest-palindromic-substring/),

**AVL Tree:**

1. [AVL Tree | Set 1 (Insertion)](http://www.geeksforgeeks.org/avl-tree-set-1-insertion/),
2. [AVL Tree | Set 2 (Deletion)](http://www.geeksforgeeks.org/avl-tree-set-2-deletion/).

***Splay Tree:***

1. [Splay Tree | Set 1 (Search)](http://www.geeksforgeeks.org/splay-tree-set-1-insert/),
2. [Splay Tree | Set 2 (Insert)](http://www.geeksforgeeks.org/splay-tree-set-2-insert-delete/)

***B Tree:***

1. [B-Tree | Set 1 (Introduction)](http://www.geeksforgeeks.org/b-tree-set-1-introduction-2/),
2. [B-Tree | Set 2 (Insert)](http://www.geeksforgeeks.org/b-tree-set-1-insert-2/),
3. [B-Tree | Set 3 (Delete)](http://www.geeksforgeeks.org/b-tree-set-3delete/)

***Segment Tree:***

1. [Segment Tree | Set 1 (Sum of given range)](http://www.geeksforgeeks.org/segment-tree-set-1-sum-of-given-range/),
2. [Segment Tree | Set 2 (Range Minimum Query)](http://www.geeksforgeeks.org/segment-tree-set-1-range-minimum-query/)

***Red-Black Tree:***

1. [Red-Black Tree Introduction](http://www.geeksforgeeks.org/red-black-tree-set-1-introduction-2/),
2. [Red Black Tree Insertion.](http://www.geeksforgeeks.org/red-black-tree-set-2-insert/)
3. [Red-Black Tree Deletion](http://www.geeksforgeeks.org/red-black-tree-set-3-delete-2/),
4. [Program for Red Black Tree Insertion](http://geeksquiz.com/c-program-red-black-tree-insertion/)

**Others:**

1. [Ternary Search Tree](http://www.geeksforgeeks.org/ternary-search-tree/),
2. [Interval Tree](http://www.geeksforgeeks.org/interval-tree/),
3. [Implement LRU Cache](http://www.geeksforgeeks.org/implement-lru-cache/),
4. [Sort numbers stored on different machines](http://www.geeksforgeeks.org/sort-numbers-stored-on-different-machines/),
5. [Find the k most frequent words from a file](http://www.geeksforgeeks.org/find-the-k-most-frequent-words-from-a-file/),
6. [Given a sequence of words, print all anagrams together](http://www.geeksforgeeks.org/given-a-sequence-of-words-print-all-anagrams-together-set-2/),
7. [Tournament Tree (Winner Tree) and Binary Heap](http://www.geeksforgeeks.org/tournament-tree-and-binary-heap/),
8. [Decision Trees – Fake (Counterfeit) Coin Puzzle (12 Coin Puzzle)](http://www.geeksforgeeks.org/decision-trees-fake-coin-puzzle/),
9. [Spaghetti Stack](http://www.geeksforgeeks.org/g-fact-87/),
10. [Data Structure for Dictionary and Spell Checker?](http://www.geeksforgeeks.org/data-structure-dictionary-spell-checker/),
11. [KD Tree](http://www.geeksforgeeks.org/k-dimensional-tree/),
12. [Binomial Heap](http://www.geeksforgeeks.org/binomial-heap-2/),
13. [KD Tree](http://www.geeksforgeeks.org/k-dimensional-tree/),
14. [Binary Indexed Tree](http://www.geeksforgeeks.org/binary-indexed-tree-or-fenwick-tree-2/).

**Array:**

* [Given an array A[] and a number x, check for pair in A[] with sum as x](http://www.geeksforgeeks.org/write-a-c-program-that-given-a-set-a-of-n-numbers-and-another-number-x-determines-whether-or-not-there-exist-two-elements-in-s-whose-sum-is-exactly-x/),
* [Majority Element](http://www.geeksforgeeks.org/majority-element/),
* [Find the Number Occurring Odd Number of Times](http://www.geeksforgeeks.org/find-the-number-occurring-odd-number-of-times/),
* [Largest Sum Contiguous Subarray](http://www.geeksforgeeks.org/largest-sum-contiguous-subarray/),
* [Find the Missing Number](http://www.geeksforgeeks.org/find-the-missing-number/),
* [Search an element in a sorted and pivoted array](http://www.geeksforgeeks.org/search-an-element-in-a-sorted-and-pivoted-array/),
* [Merge an array of size n into another array of size m+n](http://www.geeksforgeeks.org/merge-one-array-of-size-n-into-another-one-of-size-mn/),
* [Median of two sorted arrays](http://www.geeksforgeeks.org/median-of-two-sorted-arrays/),
* [Write a program to reverse an array](http://www.geeksforgeeks.org/write-a-program-to-reverse-an-array/),
* [Program for array rotation](http://www.geeksforgeeks.org/array-rotation/),
* [Reversal algorithm for array rotation](http://www.geeksforgeeks.org/program-for-array-rotation-continued-reversal-algorithm/),
* [Block swap algorithm for array rotation](http://www.geeksforgeeks.org/block-swap-algorithm-for-array-rotation/),
* [Maximum sum such that no two elements are adjacent](http://www.geeksforgeeks.org/maximum-sum-such-that-no-two-elements-are-adjacent/),
* [Leaders in an array](http://www.geeksforgeeks.org/leaders-in-an-array/),
* [Sort elements by frequency | Set 1](http://www.geeksforgeeks.org/sort-elements-by-frequency/),

1. [Count Inversions in an array](http://www.geeksforgeeks.org/counting-inversions/),
2. [Two elements whose sum is closest to zero](http://www.geeksforgeeks.org/two-elements-whose-sum-is-closest-to-zero/),
3. [Find the smallest and second smallest element in an array](http://www.geeksforgeeks.org/to-find-smallest-and-second-smallest-element-in-an-array/),
4. [Check for Majority Element in a sorted array](http://www.geeksforgeeks.org/check-for-majority-element-in-a-sorted-array/),
5. [Maximum and minimum of an array using minimum number of comparisons](http://www.geeksforgeeks.org/maximum-and-minimum-in-an-array/),
6. [Segregate 0s and 1s in an array](http://www.geeksforgeeks.org/segregate-0s-and-1s-in-an-array-by-traversing-array-once/),
7. [k largest(or smallest) elements in an array | added Min Heap method](http://www.geeksforgeeks.org/k-largestor-smallest-elements-in-an-array/),
8. [Maximum difference between two elements](http://www.geeksforgeeks.org/maximum-difference-between-two-elements/),
9. [Union and Intersection of two sorted arrays](http://www.geeksforgeeks.org/union-and-intersection-of-two-sorted-arrays-2/),
10. [Floor and Ceiling in a sorted array](http://www.geeksforgeeks.org/search-floor-and-ceil-in-a-sorted-array/),
11. [A Product Array Puzzle](http://www.geeksforgeeks.org/a-product-array-puzzle/),
12. [Segregate Even and Odd numbers](http://www.geeksforgeeks.org/segregate-even-and-odd-numbers/),
13. [Find the two repeating elements in a given array](http://www.geeksforgeeks.org/find-the-two-repeating-elements-in-a-given-array/),
14. [Sort an array of 0s, 1s and 2s](http://www.geeksforgeeks.org/sort-an-array-of-0s-1s-and-2s/),
15. [Find the Minimum length Unsorted Subarray, sorting which makes the complete array sorted](http://www.geeksforgeeks.org/minimum-length-unsorted-subarray-sorting-which-makes-the-complete-array-sorted/),
16. [Find duplicates in O(n) time and O(1) extra space](http://www.geeksforgeeks.org/find-duplicates-in-on-time-and-constant-extra-space/),
17. [Equilibrium index of an array](http://www.geeksforgeeks.org/equilibrium-index-of-an-array/),
18. [Linked List vs Array](http://www.geeksforgeeks.org/linked-list-vs-array/),
19. [Which sorting algorithm makes minimum number of memory writes?](http://www.geeksforgeeks.org/which-sorting-algorithm-makes-minimum-number-of-writes/),
20. [Turn an image by 90 degree](http://www.geeksforgeeks.org/turn-an-image-by-90-degree/),
21. [Next Greater Element](http://www.geeksforgeeks.org/next-greater-element/),
22. [Check if array elements are consecutive | Added Method 3](http://www.geeksforgeeks.org/check-if-array-elements-are-consecutive/),
23. [Find the smallest missing number](http://www.geeksforgeeks.org/find-the-first-missing-number/),
24. [Count the number of occurrences in a sorted array](http://www.geeksforgeeks.org/count-number-of-occurrences-in-a-sorted-array/),
25. [Interpolation search vs Binary search](http://www.geeksforgeeks.org/g-fact-84/),
26. [Given an array arr[], find the maximum j – i such that arr[j] > arr[i]](http://www.geeksforgeeks.org/given-an-array-arr-find-the-maximum-j-i-such-that-arrj-arri/),
27. [Maximum of all subarrays of size k (Added a O(n) method)](http://www.geeksforgeeks.org/maximum-of-all-subarrays-of-size-k/),
28. [Find whether an array is subset of another array | Added Method 3](http://www.geeksforgeeks.org/find-whether-an-array-is-subset-of-another-array-set-1/),
29. [Find the minimum distance between two numbers](http://www.geeksforgeeks.org/find-the-minimum-distance-between-two-numbers/),
30. [Find the repeating and the missing | Added 3 new methods](http://www.geeksforgeeks.org/find-a-repeating-and-a-missing-number/),
31. [Median in a stream of integers (running integers)](http://www.geeksforgeeks.org/median-of-stream-of-integers-running-integers/),
32. [Find a Fixed Point in a given array](http://www.geeksforgeeks.org/find-a-fixed-point-in-a-given-array/),
33. [Maximum Length Bitonic Subarray](http://www.geeksforgeeks.org/maximum-length-bitonic-subarray/),
34. [Find the maximum element in an array which is first increasing and then decreasing](http://www.geeksforgeeks.org/find-the-maximum-element-in-an-array-which-is-first-increasing-and-then-decreasing/),
35. [Count smaller elements on right side](http://www.geeksforgeeks.org/count-smaller-elements-on-right-side/),
36. [Minimum number of jumps to reach end](http://www.geeksforgeeks.org/minimum-number-of-jumps-to-reach-end-of-a-given-array/),
37. [Implement two stacks in an array](http://www.geeksforgeeks.org/implement-two-stacks-in-an-array/),
38. [Find subarray with given sum](http://www.geeksforgeeks.org/find-subarray-with-given-sum/),[Dynamic Programming | Set 14 (Maximum Sum Increasing Subsequence)](http://www.geeksforgeeks.org/dynamic-programming-set-14-maximum-sum-increasing-subsequence/),
39. [Longest Monotonically Increasing Subsequence Size (N log N)](http://www.geeksforgeeks.org/longest-monotonically-increasing-subsequence-size-n-log-n/),
40. [Find a triplet that sum to a given value](http://www.geeksforgeeks.org/find-a-triplet-that-sum-to-a-given-value/),
41. [Find the smallest positive number missing from an unsorted array](http://www.geeksforgeeks.org/find-the-smallest-positive-number-missing-from-an-unsorted-array/),
42. [Find the two numbers with odd occurrences in an unsorted array](http://www.geeksforgeeks.org/find-the-two-numbers-with-odd-occurences-in-an-unsorted-array/),
43. [The Celebrity Problem](http://www.geeksforgeeks.org/the-celebrity-problem/),
44. [Dynamic Programming | Set 15 (Longest Bitonic Subsequence)](http://www.geeksforgeeks.org/dynamic-programming-set-15-longest-bitonic-subsequence/),
45. [Find a sorted subsequence of size 3 in linear time](http://www.geeksforgeeks.org/find-a-sorted-subsequence-of-size-3-in-linear-time/),
46. [Largest subarray with equal number of 0s and 1s](http://www.geeksforgeeks.org/largest-subarray-with-equal-number-of-0s-and-1s/),
47. [Dynamic Programming | Set 18 (Partition problem)](http://www.geeksforgeeks.org/dynamic-programming-set-18-partition-problem/),
48. [Maximum Product Subarray](http://www.geeksforgeeks.org/maximum-product-subarray/),
49. [Find a pair with the given difference](http://www.geeksforgeeks.org/find-a-pair-with-the-given-difference/),
50. [Replace every element with the next greatest](http://www.geeksforgeeks.org/replace-every-element-with-the-greatest-on-right-side/),
51. [Dynamic Programming | Set 20 (Maximum Length Chain of Pairs)](http://www.geeksforgeeks.org/dynamic-programming-set-20-maximum-length-chain-of-pairs/),
52. [Find four elements that sum to a given value | Set 1 (n^3 solution)](http://www.geeksforgeeks.org/find-four-numbers-with-sum-equal-to-given-sum/),
53. [Find four elements that sum to a given value | Set 2 ( O(n^2Logn) Solution)](http://www.geeksforgeeks.org/find-four-elements-that-sum-to-a-given-value-set-2/),
54. [Sort a nearly sorted (or K sorted) array](http://www.geeksforgeeks.org/nearly-sorted-algorithm/),
55. [Maximum circular subarray sum](http://www.geeksforgeeks.org/maximum-contiguous-circular-sum/),
56. [Find the row with maximum number of 1s](http://www.geeksforgeeks.org/find-the-row-with-maximum-number-1s/),
57. [Median of two sorted arrays of different sizes](http://www.geeksforgeeks.org/median-of-two-sorted-arrays-of-different-sizes/),
58. [Shuffle a given array](http://www.geeksforgeeks.org/shuffle-a-given-array/),
59. [Count the number of possible triangles](http://www.geeksforgeeks.org/find-number-of-triangles-possible/),
60. [Iterative Quick Sort](http://www.geeksforgeeks.org/iterative-quick-sort/),
61. [Find the number of islands](http://www.geeksforgeeks.org/find-number-of-islands/),
62. [Construction of Longest Monotonically Increasing Subsequence (N log N)](http://www.geeksforgeeks.org/construction-of-longest-monotonically-increasing-subsequence-n-log-n/),
63. [Find the first circular tour that visits all petrol pumps](http://www.geeksforgeeks.org/find-a-tour-that-visits-all-stations/),
64. [Arrange given numbers to form the biggest number](http://www.geeksforgeeks.org/given-an-array-of-numbers-arrange-the-numbers-to-form-the-biggest-number/),
65. [Pancake sorting](http://www.geeksforgeeks.org/pancake-sorting/),
66. [A Pancake Sorting Problem](http://www.geeksforgeeks.org/a-pancake-sorting-question/),
67. [Tug of War](http://www.geeksforgeeks.org/tug-of-war/),
68. [Divide and Conquer | Set 3 (Maximum Subarray Sum)](http://www.geeksforgeeks.org/divide-and-conquer-maximum-sum-subarray/),
69. [Counting Sort](http://www.geeksforgeeks.org/counting-sort/),
70. [Merge Overlapping Intervals](http://www.geeksforgeeks.org/merging-intervals/),
71. [Find the maximum repeating number in O(n) time and O(1) extra space](http://www.geeksforgeeks.org/find-the-maximum-repeating-number-in-ok-time/),
72. [Stock Buy Sell to Maximize Profit](http://www.geeksforgeeks.org/stock-buy-sell/),
73. [Rearrange positive and negative numbers in O(n) time and O(1) extra space](http://www.geeksforgeeks.org/rearrange-positive-and-negative-numbers-publish/),
74. [Sort elements by frequency | Set 2](http://www.geeksforgeeks.org/sort-elements-by-frequency-set-2/),
75. [Find a peak element](http://www.geeksforgeeks.org/find-a-peak-in-a-given-array/),
76. [Print all possible combinations of r elements in a given array of size n](http://www.geeksforgeeks.org/print-all-possible-combinations-of-r-elements-in-a-given-array-of-size-n/),
77. [Given an array of of size n and a number k, find all elements that appear more than n/k times](http://www.geeksforgeeks.org/given-an-array-of-of-size-n-finds-all-the-elements-that-appear-more-than-nk-times/),
78. [Find the point where a monotonically increasing function becomes positive first time](http://www.geeksforgeeks.org/find-the-point-where-a-function-becomes-negative/),
79. [Find the Increasing subsequence of length three with maximum product](http://www.geeksforgeeks.org/increasing-subsequence-of-length-three-with-maximum-product/),
80. [Find the minimum element in a sorted and rotated array](http://www.geeksforgeeks.org/find-minimum-element-in-a-sorted-and-rotated-array/),
81. [Stable Marriage Problem](http://www.geeksforgeeks.org/stable-marriage-problem/),
82. [Merge k sorted arrays | Set 1](http://www.geeksforgeeks.org/merge-k-sorted-arrays/),[Radix Sort](http://www.geeksforgeeks.org/radix-sort/),
83. [Move all zeroes to end of array](http://www.geeksforgeeks.org/move-zeroes-end-array/),
84. [Find number of pairs such that x^y > y^x](http://www.geeksforgeeks.org/find-number-pairs-xy-yx/),
85. [Count all distinct pairs with difference equal to k](http://www.geeksforgeeks.org/count-pairs-difference-equal-k/),
86. [Find if there is a subarray with 0 sum](http://www.geeksforgeeks.org/find-if-there-is-a-subarray-with-0-sum/),[Smallest subarray with sum greater than a given value](http://www.geeksforgeeks.org/minimum-length-subarray-sum-greater-given-value/),
87. [Sort an array according to the order defined by another array](http://www.geeksforgeeks.org/sort-array-according-order-defined-another-array/),
88. [Maximum Sum Path in Two Arrays](http://www.geeksforgeeks.org/maximum-sum-path-across-two-arrays/),
89. [Check if a given array contains duplicate elements within k distance from each other](http://www.geeksforgeeks.org/check-given-array-contains-duplicate-elements-within-k-distance/),
90. [Sort an array in wave form](http://www.geeksforgeeks.org/sort-array-wave-form-2/),
91. [K’th Smallest/Largest Element in Unsorted Array, K’th Smallest/Largest Element in Unsorted Array in Expected Linear Time,](http://www.geeksforgeeks.org/find-index-0-replaced-1-get-longest-continuous-sequence-1s-binary-array/)
92. [K’th Smallest/Largest Element in Unsorted Array in Worst Case Linear Time](http://www.geeksforgeeks.org/kth-smallestlargest-element-unsorted-array-set-3-worst-case-linear-time/),
93. [Find Index of 0 to be replaced with 1 to get longest continuous sequence of 1s in a binary array](http://www.geeksforgeeks.org/find-index-0-replaced-1-get-longest-continuous-sequence-1s-binary-array/),
94. [Find the closest pair from two sorted arrays](http://www.geeksforgeeks.org/given-two-sorted-arrays-number-x-find-pair-whose-sum-closest-x/),
95. [Given a sorted array and a number x, find the pair in array whose sum is closest to x](http://geeksquiz.com/given-sorted-array-number-x-find-pair-array-whose-sum-closest-x/),
96. [Count 1’s in a sorted binary array](http://geeksquiz.com/count-1s-sorted-binary-array/),
97. [Print All Distinct Elements of a given integer array](http://geeksquiz.com/print-distinct-elements-given-integer-array/)[Construct an array from its pair-sum array](http://geeksquiz.com/construct-array-pair-sum-array/),
98. [Find common elements in three sorted arrays](http://www.geeksforgeeks.org/find-common-elements-three-sorted-arrays/),
99. [Find the first repeating element in an array of integers](http://www.geeksforgeeks.org/find-first-repeating-element-array-integers/),
100. [Find the smallest positive integer value that cannot be represented as sum of any subset of a given array](http://www.geeksforgeeks.org/find-smallest-value-represented-sum-subset-given-array/),
101. [Rearrange an array such that ‘arr[j]’ becomes ‘i’ if ‘arr[i]’ is ‘j’](http://www.geeksforgeeks.org/rearrange-array-arrj-becomes-arri-j/),
102. [Find position of an element in a sorted array of infinite numbers](http://www.geeksforgeeks.org/find-position-element-sorted-array-infinite-numbers/),
103. [Can QuickSort be implemented in O(nLogn) worst case time complexity?](http://www.geeksforgeeks.org/can-quicksort-implemented-onlogn-worst-case-time-complexity/),
104. [Check if a given array contains duplicate elements within k distance from each other](http://www.geeksforgeeks.org/check-given-array-contains-duplicate-elements-within-k-distance/)
105. [Quiz on Array](http://geeksquiz.com/data-structure/array/)

**Matrix:**

1. [Search in a row wise and column wise sorted matrix](http://www.geeksforgeeks.org/search-in-row-wise-and-column-wise-sorted-matrix/),
2. [Print a given matrix in spiral form](http://www.geeksforgeeks.org/print-a-given-matrix-in-spiral-form/),
3. [A Boolean Matrix Question](http://www.geeksforgeeks.org/a-boolean-matrix-question/),
4. [Print unique rows in a given boolean matrix](http://www.geeksforgeeks.org/print-unique-rows/),
5. [Maximum size square sub-matrix with all 1s](http://www.geeksforgeeks.org/maximum-size-sub-matrix-with-all-1s-in-a-binary-matrix/),
6. [Print unique rows in a given boolean matrix](http://www.geeksforgeeks.org/print-unique-rows/),
7. [Inplace M x N size matrix transpose | Updated](http://www.geeksforgeeks.org/inplace-m-x-n-size-matrix-transpose/),
8. [Print Matrix Diagonally](http://www.geeksforgeeks.org/print-matrix-diagonally/),
9. [Dynamic Programming | Set 27 (Maximum sum rectangle in a 2D matrix)](http://www.geeksforgeeks.org/dynamic-programming-set-27-max-sum-rectangle-in-a-2d-matrix/),
10. [Strassen’s Matrix Multiplication](http://www.geeksforgeeks.org/strassens-matrix-multiplication/),
11. [Create a matrix with alternating rectangles of O and X](http://www.geeksforgeeks.org/create-a-matrix-with-alternating-rectangles-of-0-and-x/),
12. [Find the row with maximum number of 1s](http://www.geeksforgeeks.org/find-the-row-with-maximum-number-1s/),
13. [Print all elements in sorted order from row and column wise sorted matrix](http://www.geeksforgeeks.org/print-elements-sorted-order-row-column-wise-sorted-matrix/),
14. [Given an n x n square matrix, find sum of all sub-squares of size k x k](http://www.geeksforgeeks.org/given-n-x-n-square-matrix-find-sum-sub-squares-size-k-x-k/),
15. [Count number of islands where every island is row-wise and column-wise separated](http://www.geeksforgeeks.org/count-number-islands-every-island-separated-line/),
16. [Find a common element in all rows of a given row-wise sorted matrix](http://www.geeksforgeeks.org/find-common-element-rows-row-wise-sorted-matrix/),
17. [Given a matrix of ‘O’ and ‘X’, replace ‘O’ with ‘X’ if surrounded by ‘X’](http://www.geeksforgeeks.org/given-matrix-o-x-replace-o-x-surrounded-x/)

**Misc:**

[Commonly Asked Data Structure Interview Questions | Set 1](http://geeksquiz.com/commonly-asked-data-structure-interview-questions-set-1/),

[A data structure for n elements and O(1) operations](http://geeksquiz.com/data-structure-n-elements-o1-operations/)