**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*DS-Topics\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

***Singly Linked List:***

1. [Introduction to Linked List](http://geeksquiz.com/linked-list-set-1-introduction/)& [Linked List vs Array](https://www.geeksforgeeks.org/linked-list-vs-array/):

**Advantages over arrays  
1)**Dynamic size  
**2)** Ease of insertion/deletion

**Drawbacks:**  
**1)** Random access is not allowed. We have to access elements sequentially starting from the first node. So we cannot do binary search with linked lists efficiently with its default implementation.  
**2)** Extra memory space for a pointer is required with each element of the list.

1. [Linked List Insertion](http://geeksquiz.com/linked-list-set-2-inserting-a-node/), [Linked List Deletion (Deleting a given key)](http://geeksquiz.com/linked-list-set-3-deleting-node/): have programs
2. [Find Length of a Linked List (Iterative and Recursive)](http://geeksquiz.com/find-length-of-a-linked-list-iterative-and-recursive/):

int linkedlist::getCount()

{

    int count = 0; // Initialize count

    Node\* temp = head; // Initialize current

    while (temp != NULL)

    {

        count++;

        temp = temp->next;

    }

    return count;

}

1. [How to write C functions that modify head pointer of a Linked List?](https://www.geeksforgeeks.org/how-to-write-functions-that-modify-the-head-pointer-of-a-linked-list/)--> will be modified if we add node at beginning or delete from beginning.
2. [Swap nodes in a linked list without swapping data](https://www.geeksforgeeks.org/swap-nodes-in-a-linked-list-without-swapping-data/)--->remaining
3. [Reverse a linked list](https://www.geeksforgeeks.org/write-a-function-to-reverse-the-nodes-of-a-linked-list/)- have code
4. [Merge two sorted linked lists](https://www.geeksforgeeks.org/merge-two-sorted-linked-lists/)

Node\* SortedMerge(Node\* a, Node\* b)

{

    Node\* result = NULL;

    /\* Base cases \*/

    if (a == NULL)

         return(b);

    else if (b == NULL)

        return(a);

    /\* Pick either a or b, and recur \*/

    if (a->data <= b->data)

    {

        result = a;

        result->next = SortedMerge(a->next, b);

    }

    else

    {

        result = b;

        result->next = SortedMerge(a, b->next);

    }

    return(result);

}

1. [Merge Sort for Linked Lists](https://www.geeksforgeeks.org/merge-sort-for-linked-list/)
2. [Reverse a Linked List in groups of given size](https://www.geeksforgeeks.org/reverse-a-list-in-groups-of-given-size/)
3. [Detect and Remove Loop in a Linked List](https://www.geeksforgeeks.org/detect-and-remove-loop-in-a-linked-list/)
4. [Add two numbers represented by linked lists | Set 1](https://www.geeksforgeeks.org/add-two-numbers-represented-by-linked-lists/)
5. [Rotate a Linked List](https://www.geeksforgeeks.org/rotate-a-linked-list/)
6. [Generic Linked List in C](https://www.geeksforgeeks.org/generic-linked-list-in-c-2/)

***Circular Linked List:***

1. [Circular Linked List Introduction and Applications,](http://geeksquiz.com/circular-linked-list/)
2. [Circular Singly Linked List Insertion<](https://www.geeksforgeeks.org/circular-singly-linked-list-insertion/)
3. [Circular Linked List Traversal](http://geeksquiz.com/circular-linked-list-set-2-traversal/)
4. [Split a Circular Linked List into two halves](https://www.geeksforgeeks.org/split-a-circular-linked-list-into-two-halves/)
5. [Sorted insert for circular linked list](https://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](https://www.geeksforgeeks.org/delete-a-node-in-a-doubly-linked-list/)
3. [Reverse a Doubly Linked List](https://www.geeksforgeeks.org/reverse-a-doubly-linked-list/)
4. [The Great Tree-List Recursion Problem.](https://www.geeksforgeeks.org/the-great-tree-list-recursion-problem/)
5. [QuickSort on Doubly Linked List](https://www.geeksforgeeks.org/quicksort-for-linked-list/)
6. [Merge Sort for Doubly Linked List](https://www.geeksforgeeks.org/merge-sort-for-doubly-linked-list/)

[**All Articles of Linked List**](https://www.geeksforgeeks.org/data-structures/linked-list/)  
[Quiz on Linked List](http://geeksquiz.com/data-structure/linked-list/)  
[Coding Practice on Linked List](https://practice.geeksforgeeks.org/topics/Linked-List/)  
[Recent Articles on Linked List](https://www.geeksforgeeks.org/category/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 String using Stack](http://geeksquiz.com/stack-set-3-reverse-string-using-stack/)
5. [Implement two stacks in an array](https://www.geeksforgeeks.org/implement-two-stacks-in-an-array/)
6. [Check for balanced parentheses in an expression](https://www.geeksforgeeks.org/check-for-balanced-parentheses-in-an-expression/)
7. [Next Greater Element](https://www.geeksforgeeks.org/next-greater-element/)
8. [Reverse a stack using recursion](https://www.geeksforgeeks.org/reverse-a-stack-using-recursion/)
9. [Sort a stack using recursion](https://www.geeksforgeeks.org/sort-a-stack-using-recursion/)
10. [The Stock Span Problem](https://www.geeksforgeeks.org/the-stock-span-problem/)
11. [Design and Implement Special Stack Data Structure](https://www.geeksforgeeks.org/design-and-implement-special-stack-data-structure/)
12. [Implement Stack using Queues](https://www.geeksforgeeks.org/implement-stack-using-queue/)
13. [Design a stack with operations on middle element](https://www.geeksforgeeks.org/design-a-stack-with-find-middle-operation/)
14. [How to efficiently implement k stacks in a single array?](https://www.geeksforgeeks.org/efficiently-implement-k-stacks-single-array/)
15. [Sort a stack using recursion](https://www.geeksforgeeks.org/sort-a-stack-using-recursion/)

[Quiz on Stack](http://geeksquiz.com/data-structure/stack/)  
[**All Articles on Stack**](https://www.geeksforgeeks.org/stack/)  
[Coding Practice on Stack](https://practice.geeksforgeeks.org/topics/Stack/)  
[Recent Articles on Stack](https://www.geeksforgeeks.org/category/stack/)

**Queue:**

1. [Queue Introduction and Array Implementation](http://geeksquiz.com/queue-set-1introduction-and-array-implementation/)
2. [Linked List Implementation of Queue](http://geeksquiz.com/queue-set-2-linked-list-implementation/)
3. [Applications of Queue Data Structure](https://www.geeksforgeeks.org/applications-of-queue-data-structure/)
4. [Priority Queue Introduction](http://geeksquiz.com/priority-queue-set-1-introduction/)
5. [Deque (Introduction and Applications)](http://geeksquiz.com/deque-set-1-introduction-applications/)
6. [Implementation of Deque using circular array](https://www.geeksforgeeks.org/implementation-deque-using-circular-array/)
7. [Implement Queue using Stacks](https://www.geeksforgeeks.org/queue-using-stacks/)
8. [Find the first circular tour that visits all petrol pumps](https://www.geeksforgeeks.org/find-a-tour-that-visits-all-stations/)
9. [Maximum of all subarrays of size k](https://www.geeksforgeeks.org/maximum-of-all-subarrays-of-size-k/)
10. [An Interesting Method to Generate Binary Numbers from 1 to n](https://www.geeksforgeeks.org/interesting-method-generate-binary-numbers-1-n/)
11. [How to efficiently implement k Queues in a single array?](https://www.geeksforgeeks.org/efficiently-implement-k-queues-single-array/)

[Quiz on Queue](http://geeksquiz.com/data-structure/queue/)  
[**All Articles on Queue**](https://www.geeksforgeeks.org/queue/)  
[Coding Practice on Queue](https://practice.geeksforgeeks.org/topics/Queue/)  
[Recent Articles on Queue](https://www.geeksforgeeks.org/category/queue/)

**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. [Handshaking Lemma and Interesting Tree Properties](https://www.geeksforgeeks.org/handshaking-lemma-and-interesting-tree-properties/)
5. [Enumeration of Binary Tree](http://geeksquiz.com/enumeration-of-binary-trees/)
6. [Applications of tree data structure](https://www.geeksforgeeks.org/applications-of-tree-data-structure/)
7. [Tree Traversals](https://www.geeksforgeeks.org/618/)
8. [BFS vs DFS for Binary Tree](https://www.geeksforgeeks.org/bfs-vs-dfs-binary-tree/)
9. [Level Order Tree Traversal](https://www.geeksforgeeks.org/level-order-tree-traversal/)
10. [Diameter of a Binary Tree](https://www.geeksforgeeks.org/diameter-of-a-binary-tree/)
11. [Inorder Tree Traversal without Recursion](https://www.geeksforgeeks.org/inorder-tree-traversal-without-recursion/)
12. [Inorder Tree Traversal without recursion and without stack!](https://www.geeksforgeeks.org/inorder-tree-traversal-without-recursion-and-without-stack/)
13. [Threaded Binary Tree](http://geeksquiz.com/threaded-binary-tree/)
14. [Maximum Depth or Height of a Tree](https://www.geeksforgeeks.org/write-a-c-program-to-find-the-maximum-depth-or-height-of-a-tree/)
15. [If you are given two traversal sequences, can you construct the binary tree?](https://www.geeksforgeeks.org/if-you-are-given-two-traversal-sequences-can-you-construct-the-binary-tree/)
16. [Clone a Binary Tree with Random Pointers](https://www.geeksforgeeks.org/clone-binary-tree-random-pointers/)
17. [Construct Tree from given Inorder and Preorder traversals](https://www.geeksforgeeks.org/construct-tree-from-given-inorder-and-preorder-traversal/)
18. [Maximum width of a binary tree](https://www.geeksforgeeks.org/maximum-width-of-a-binary-tree/)
19. [Print nodes at k distance from root](https://www.geeksforgeeks.org/print-nodes-at-k-distance-from-root/)
20. [Print Ancestors of a given node in Binary Tree](https://www.geeksforgeeks.org/print-ancestors-of-a-given-node-in-binary-tree/)
21. [Check if a binary tree is subtree of another binary tree](https://www.geeksforgeeks.org/check-if-a-binary-tree-is-subtree-of-another-binary-tree/)
22. [Connect nodes at same level](https://www.geeksforgeeks.org/connect-nodes-at-same-level/)

[Quiz on Binary Tree](http://geeksquiz.com/data-structure/binary-trees/)  
[Quiz on Binary Tree Traversals](http://geeksquiz.com/data-structure/tree-traversals/)  
[**All articles on Binary Tree**](https://www.geeksforgeeks.org/binary-tree-2/)  
[Coding Practice on Binary Tree](https://practice.geeksforgeeks.org/topics/Tree/)  
[Recent Articles on Tree](https://www.geeksforgeeks.org/category/tree/)

**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](https://www.geeksforgeeks.org/find-the-minimum-element-in-a-binary-search-tree/)
4. [Inorder predecessor and successor for a given key in BST](https://www.geeksforgeeks.org/inorder-predecessor-successor-given-key-bst/)
5. [Check if a binary tree is BST or not](https://www.geeksforgeeks.org/a-program-to-check-if-a-binary-tree-is-bst-or-not/)
6. [Lowest Common Ancestor in a Binary Search Tree.](https://www.geeksforgeeks.org/lowest-common-ancestor-in-a-binary-search-tree/)
7. [Inorder Successor in Binary Search Tree](https://www.geeksforgeeks.org/inorder-successor-in-binary-search-tree/)
8. [Find k-th smallest element in BST (Order Statistics in BST)](https://www.geeksforgeeks.org/find-k-th-smallest-element-in-bst-order-statistics-in-bst/)
9. [Merge two BSTs with limited extra space](https://www.geeksforgeeks.org/merge-two-bsts-with-limited-extra-space/)
10. [Two nodes of a BST are swapped, correct the BST](https://www.geeksforgeeks.org/fix-two-swapped-nodes-of-bst/)
11. [Floor and Ceil from a BST](https://www.geeksforgeeks.org/floor-and-ceil-from-a-bst/)
12. [In-place conversion of Sorted DLL to Balanced BST](https://www.geeksforgeeks.org/in-place-conversion-of-sorted-dll-to-balanced-bst/)
13. [Find a pair with given sum in a Balanced BST](https://www.geeksforgeeks.org/find-a-pair-with-given-sum-in-bst/)
14. [Total number of possible Binary Search Trees with n keys](https://www.geeksforgeeks.org/total-number-of-possible-binary-search-trees-with-n-keys/)
15. [Merge Two Balanced Binary Search Trees](https://www.geeksforgeeks.org/merge-two-balanced-binary-search-trees/)
16. [Binary Tree to Binary Search Tree Conversion](https://www.geeksforgeeks.org/binary-tree-to-binary-search-tree-conversion/)

[Quiz on Binary Search Trees](http://geeksquiz.com/data-structure/binary-search-trees/)  
[Quiz on Balanced Binary Search Trees](http://geeksquiz.com/data-structure/balanced-binary-search-trees/)  
[**All Articles on Binary Search Tree**](https://www.geeksforgeeks.org/binary-search-tree/)  
[Coding Practice on Binary Search Tree](https://practice.geeksforgeeks.org/topics/BST/)  
[Recent Articles on BST](https://www.geeksforgeeks.org/category/binary-search-tree/)

**Heap:**

1. [Binary Heap](http://geeksquiz.com/binary-heap/)
2. [Why is Binary Heap Preferred over BST for Priority Queue?](https://www.geeksforgeeks.org/why-is-binary-heap-preferred-over-bst-for-priority-queue/)
3. [Binomial Heap](https://www.geeksforgeeks.org/binomial-heap-2/)
4. [Fibonacci Heap](https://www.geeksforgeeks.org/fibonacci-heap-set-1-introduction/)
5. [Heap Sort](http://geeksquiz.com/heap-sort/)
6. [K’th Largest Element in an array](https://www.geeksforgeeks.org/k-largestor-smallest-elements-in-an-array/)
7. [Sort an almost sorted array/](https://www.geeksforgeeks.org/nearly-sorted-algorithm/)
8. [Tournament Tree (Winner Tree) and Binary Heap](https://www.geeksforgeeks.org/tournament-tree-and-binary-heap/)

[**All Articles on Heap**](https://www.geeksforgeeks.org/heap/)  
[Quiz on Heap](http://geeksquiz.com/data-structure/heap/)  
[Coding Practice on Heap](https://practice.geeksforgeeks.org/tag-page.php?tag=heap&isCmp=0)  
[Recent Articles on Heap](https://www.geeksforgeeks.org/category/heap/)

**Hashing:**

1. [Hashing Introduction](http://geeksquiz.com/hashing-set-1-introduction/)
2. [Separate Chaining for Collision Handling](http://geeksquiz.com/hashing-set-2-separate-chaining/)
3. [Open Addressing for Collision Handling](http://geeksquiz.com/hashing-set-3-open-addressing/)
4. [Print a Binary Tree in Vertical Order](https://www.geeksforgeeks.org/print-binary-tree-vertical-order-set-2/)
5. [Find whether an array is subset of another array](https://www.geeksforgeeks.org/find-whether-an-array-is-subset-of-another-array-set-1/)
6. [Union and Intersection of two Linked Lists](https://www.geeksforgeeks.org/union-and-intersection-of-two-linked-lists/)
7. [Find a pair with given sum](https://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/)
8. [Check if a given array contains duplicate elements within k distance from each other](https://www.geeksforgeeks.org/check-given-array-contains-duplicate-elements-within-k-distance/)
9. [Find Itinerary from a given list of tickets](https://www.geeksforgeeks.org/find-itinerary-from-a-given-list-of-tickets/)
10. [Find number of Employees Under every Employee](https://www.geeksforgeeks.org/find-number-of-employees-under-every-manager/)

[Quiz on Hashing](http://geeksquiz.com/data-structure/hash/)  
[**All Articles on Hashing**](https://www.geeksforgeeks.org/hashing/)  
[Coding Practice on Hashing](https://practice.geeksforgeeks.org/topics/Hashing/)  
[Recent Articles on Hashing](https://www.geeksforgeeks.org/category/hash/)

**Graph:**

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

1. [Graph and its representations](https://www.geeksforgeeks.org/graph-and-its-representations/)
2. [Breadth First Traversal for a Graph](https://www.geeksforgeeks.org/breadth-first-traversal-for-a-graph/)
3. [Depth First Traversal for a Graph](https://www.geeksforgeeks.org/depth-first-traversal-for-a-graph/)
4. [Applications of Depth First Search](https://www.geeksforgeeks.org/applications-of-depth-first-search/)
5. [Applications of Breadth First Traversal](https://www.geeksforgeeks.org/applications-of-breadth-first-traversal/)
6. [Detect Cycle in a Directed Graph](https://www.geeksforgeeks.org/detect-cycle-in-a-graph/)
7. [Detect Cycle in a an Undirected Graph](https://www.geeksforgeeks.org/union-find/)
8. [Detect cycle in an undirected graph](https://www.geeksforgeeks.org/detect-cycle-undirected-graph/)
9. [Longest Path in a Directed Acyclic Graph](https://www.geeksforgeeks.org/find-longest-path-directed-acyclic-graph/)
10. [Topological Sorting](https://www.geeksforgeeks.org/topological-sorting/)
11. [Check whether a given graph is Bipartite or not](https://www.geeksforgeeks.org/bipartite-graph/)
12. [Snake and Ladder Problem](https://www.geeksforgeeks.org/snake-ladder-problem-2/)
13. [Minimize Cash Flow among a given set of friends who have borrowed money from each other](https://www.geeksforgeeks.org/minimize-cash-flow-among-given-set-friends-borrowed-money/)
14. [Boggle (Find all possible words in a board of characters)](https://www.geeksforgeeks.org/boggle-find-possible-words-board-characters/)
15. [Assign directions to edges so that the directed graph remains acyclic](https://www.geeksforgeeks.org/assign-directions-to-edges-so-that-the-directed-graph-remains-acyclic/)

[**All Articles on Graph Data Structure**](https://www.geeksforgeeks.org/graph-data-structure-and-algorithms/)  
[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/)  
[Coding Practice on Graph](https://practice.geeksforgeeks.org/topics/Graph/)  
[Recent Articles on Graph](https://www.geeksforgeeks.org/category/graph/)

**Advanced Data Structure:**

***Advanced Lists:***

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

***Segment Tree:***

1. [Segment Tree | Set 1 (Sum of given range)](https://www.geeksforgeeks.org/segment-tree-set-1-sum-of-given-range/)
2. [Segment Tree | Set 2 (Range Minimum Query)](https://www.geeksforgeeks.org/segment-tree-set-1-range-minimum-query/)
3. [Lazy Propagation in Segment Tree](https://www.geeksforgeeks.org/lazy-propagation-in-segment-tree/)
4. [Persistent Segment Tree | Set 1 (Introduction)](https://www.geeksforgeeks.org/persistent-segment-tree-set-1-introduction/)

[**All articles on Segment Tree**](https://www.geeksforgeeks.org/tag/segment-tree/)  
***Trie:***

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

[**All Articles on Trie**](https://www.geeksforgeeks.org/tag/trie/)  
***Binary Indexed Tree:***

1. [Binary Indexed Tree](https://www.geeksforgeeks.org/binary-indexed-tree-or-fenwick-tree-2/)
2. [Two Dimensional Binary Indexed Tree or Fenwick Tree](https://www.geeksforgeeks.org/two-dimensional-binary-indexed-tree-or-fenwick-tree/)
3. [Binary Indexed Tree : Range Updates and Point Queries](https://www.geeksforgeeks.org/binary-indexed-tree-range-updates-point-queries/)
4. [Binary Indexed Tree : Range Update and Range Queries](https://www.geeksforgeeks.org/binary-indexed-tree-range-update-range-queries/)

[**All Articles on Binary Indexed Tree**](https://www.geeksforgeeks.org/tag/binary-indexed-tree/)  
***Suffix Array and Suffix Tree***:

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

[**All Articles on Suffix Tree**](https://www.geeksforgeeks.org/tag/suffix-tree/)  
**AVL Tree:**

1. [AVL Tree | Set 1 (Insertion)](https://www.geeksforgeeks.org/avl-tree-set-1-insertion/)
2. [AVL Tree | Set 2 (Deletion)](https://www.geeksforgeeks.org/avl-tree-set-2-deletion/)
3. [AVL with duplicate keys](http://geeksquiz.com/avl-with-duplicate-keys/)

***Splay Tree:***

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

***B Tree:***

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

***Red-Black Tree:***

1. [Red-Black Tree Introduction](https://www.geeksforgeeks.org/red-black-tree-set-1-introduction-2/)
2. [Red Black Tree Insertion.](https://www.geeksforgeeks.org/red-black-tree-set-2-insert/)
3. [Red-Black Tree Deletion](https://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/)

[**All Articles on Self-Balancing BSTs**](https://www.geeksforgeeks.org/tag/self-balancing-bst/)

***K Dimensional Tree:***

1. [KD Tree (Search and Insert)](https://www.geeksforgeeks.org/k-dimensional-tree/)
2. [K D Tree (Find Minimum)](https://www.geeksforgeeks.org/k-dimensional-tree-set-2-find-minimum/)
3. [K D Tree (Delete)](https://www.geeksforgeeks.org/k-dimensional-tree-set-3-delete/)

***Others:***

* 1. [Treap (A Randomized Binary Search Tree)](https://www.geeksforgeeks.org/treap-a-randomized-binary-search-tree/)
  2. [Ternary Search Tree](https://www.geeksforgeeks.org/ternary-search-tree/)
  3. [Interval Tree](https://www.geeksforgeeks.org/interval-tree/)
  4. [Implement LRU Cache](https://www.geeksforgeeks.org/implement-lru-cache/)
  5. [Sort numbers stored on different machines](https://www.geeksforgeeks.org/sort-numbers-stored-on-different-machines/)
  6. [Find the k most frequent words from a file](https://www.geeksforgeeks.org/find-the-k-most-frequent-words-from-a-file/)
  7. [Given a sequence of words, print all anagrams together](https://www.geeksforgeeks.org/given-a-sequence-of-words-print-all-anagrams-together-set-2/)
  8. [Tournament Tree (Winner Tree) and Binary Heap](https://www.geeksforgeeks.org/tournament-tree-and-binary-heap/)
  9. [Decision Trees – Fake (Counterfeit) Coin Puzzle (12 Coin Puzzle)](https://www.geeksforgeeks.org/decision-trees-fake-coin-puzzle/)
  10. [Spaghetti Stack](https://www.geeksforgeeks.org/g-fact-87/)
  11. [Data Structure for Dictionary and Spell Checker?](https://www.geeksforgeeks.org/data-structure-dictionary-spell-checker/)
  12. [Cartesian Tree](https://www.geeksforgeeks.org/cartesian-tree/)
  13. [Cartesian Tree Sorting](https://www.geeksforgeeks.org/cartesian-tree-sorting/)
  14. [Sparse Set](https://www.geeksforgeeks.org/sparse-set/)
  15. [Centroid Decomposition of Tree](https://www.geeksforgeeks.org/centroid-decomposition-of-tree/)
  16. [Gomory-Hu Tree](https://www.geeksforgeeks.org/gomory-hu-tree-introduction/)

[Recent Articles on Advanced Data Structures.](https://www.geeksforgeeks.org/category/advanced-data-structure/)

**Array:**

* + 1. [Search, insert and delete in an unsorted array](https://www.geeksforgeeks.org/search-insert-and-delete-in-an-unsorted-array/)
    2. [Search, insert and delete in a sorted array](https://www.geeksforgeeks.org/search-insert-and-delete-in-a-sorted-array/)
    3. [Write a program to reverse an array](https://www.geeksforgeeks.org/write-a-program-to-reverse-an-array/)
    4. [Leaders in an array](https://www.geeksforgeeks.org/leaders-in-an-array/)
    5. [Given an array A[] and a number x, check for pair in A[] with sum as x](https://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. [Majority Element](https://www.geeksforgeeks.org/majority-element/)
    7. [Find the Number Occurring Odd Number of Times](https://www.geeksforgeeks.org/find-the-number-occurring-odd-number-of-times/)
    8. [Largest Sum Contiguous Subarray](https://www.geeksforgeeks.org/largest-sum-contiguous-subarray/)
    9. [Find the Missing Number](https://www.geeksforgeeks.org/find-the-missing-number/)
    10. [Search an element in a sorted and pivoted array](https://www.geeksforgeeks.org/search-an-element-in-a-sorted-and-pivoted-array/)
    11. [Merge an array of size n into another array of size m+n](https://www.geeksforgeeks.org/merge-one-array-of-size-n-into-another-one-of-size-mn/)
    12. [Median of two sorted arrays](https://www.geeksforgeeks.org/median-of-two-sorted-arrays/)
    13. [Program for array rotation](https://www.geeksforgeeks.org/array-rotation/)
    14. [Reversal algorithm for array rotation](https://www.geeksforgeeks.org/program-for-array-rotation-continued-reversal-algorithm/)
    15. [Block swap algorithm for array rotation](https://www.geeksforgeeks.org/block-swap-algorithm-for-array-rotation/)
    16. [Maximum sum such that no two elements are adjacent](https://www.geeksforgeeks.org/maximum-sum-such-that-no-two-elements-are-adjacent/)
    17. [Sort elements by frequency | Set 1](https://www.geeksforgeeks.org/sort-elements-by-frequency/)
    18. [Count Inversions in an array](https://www.geeksforgeeks.org/counting-inversions/)

[**All Articles on Array**](https://www.geeksforgeeks.org/array/)  
[Coding Practice on Array](https://practice.geeksforgeeks.org/topics/Array/)  
[Quiz on Array](http://geeksquiz.com/data-structure/array/)  
[Coding Practice on Array](https://practice.geeksforgeeks.org/topics/Array/)  
[Recent Articles on Array](https://www.geeksforgeeks.org/category/c-arrays/)

**Matrix:**

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

[**All Articles on Matrix**](https://www.geeksforgeeks.org/matrix/)  
[Coding Practice on Matrix](https://practice.geeksforgeeks.org/topics/Matrix/)  
[Recent Articles on Matrix.](https://www.geeksforgeeks.org/category/matrix/)