1. **Write a program to Delete a Tree**
2. **Write a Program to Find the Maximum Depth or Height of a Tree**
3. **Write Code to Determine if Two Trees are Identical**
4. **Write a program to Calculate Size of a tree**
5. **Root to leaf path sum equal to a given number**
6. **How to determine if a binary tree is height-balanced?**
7. **Diameter of a Binary Tree**
8. **Check for Children Sum Property in a Binary Tree**
9. **Program to count leaf nodes in a binary tree**
10. The Great Tree-List Recursion Problem
11. **Given a binary tree, print out all of its root-to-leaf paths one per line**
12. **Populate Inorder Successor for all nodes**
13. Connect nodes at same level using constant extra space
14. **Connect nodes at same level**
15. **Check if a binary tree is subtree of another binary tree | Set 1**
16. **Check if a given Binary Tree is SumTree**
17. **Print Ancestors of a given node in Binary Tree**
18. **Get Level of a node in a Binary Tree**
19. **Print nodes at k distance from root**
20. Foldable Binary Trees
21. **Maximum width of a binary tree**
22. **Double Tree**
23. **Given a binary tree, print all root-to-leaf paths**
24. **Linked complete binary tree & its creation**
25. **Check whether a given Binary Tree is Complete or not | Set 1 (Iterative Solution)**
26. **Find the maximum sum leaf to root path in a Binary Tree**
27. **Vertical Sum in a given Binary Tree | Set 1**
28. **Sum of all the numbers that are formed from root to leaf paths**
29. **Find next right node of a given key**
30. **Deepest left leaf node in a binary tree**
31. **Extract Leaves of a Binary Tree in a Doubly Linked List**
32. **Remove all nodes which don’t lie in any path with sum>= k**
33. **Print Left View of a Binary Tree**
34. **Check if all leaves are at same level**
35. **Find depth of the deepest odd level leaf node**
36. **Difference between sums of odd level and even level nodes of a Binary Tree**
37. Custom Tree Problem
38. **Iterative Method to find Height of Binary Tree**
39. Tree Isomorphism Problem
40. Check if a binary tree is subtree of another binary tree | Set 2
41. Find the maximum path sum between two leaves of a binary tree
42. **Threaded Binary Tree**
43. **Reverse alternate levels of a perfect binary tree**
44. **Print Right View of a Binary Tree**
45. Print all nodes at distance k from a given node
46. **Find distance between two given keys of a Binary Tree**
47. **Print all nodes that don’t have sibling**
48. Check if a given Binary Tree is height balanced like a Red-Black Tree
49. **Print all nodes that are at distance k from a leaf node**
50. Find the closest leaf in a Binary Tree
51. **Diagonal Sum of a Binary Tree**
52. Bottom View of a Binary Tree
53. Print Nodes in Top View of Binary Tree
54. Serialize and Deserialize an N-ary Tree
55. Check if a given graph is tree or not
56. **Print nodes between two given level numbers of a binary tree**
57. Find Height of Binary Tree represented by Parent array
58. Minimum no. of iterations to pass information to all nodes in the tree
59. **Check if two nodes are cousins in a Binary Tree**
60. **Find Minimum Depth of a Binary Tree**
61. Maximum Path Sum in a Binary Tree
62. **Expression Tree**
63. **Iterative Search for a key ‘x’ in Binary Tree**
64. **Find maximum (or minimum) in Binary Tree**
65. **Find sum of all left leaves in a given Binary Tree**
66. **Remove nodes on root to leaf paths of length < K**

Longest Common Ancestor :

1. **Lowest Common Ancestor in a Binary Tree | Set 1**
2. **Find distance between two nodes of a Binary Tree**
3. **Lowest Common Ancestor in a Binary Search Tree**
4. **Print common nodes on path from root (or common ancestors)**
5. More articles on LCA

Traversals :

1. **Tree Traversals**
2. **Level Order Tree Traversal**
3. **Print level order traversal line by line | Set 1**
4. **Inorder Tree Traversal without Recursion**
5. **Inorder Tree Traversal without recursion and without stack!**
6. **Iterative Preorder Traversal**
7. **Morris traversal for Preorder**
8. Iterative Postorder Traversal | Set 1 (Using Two Stacks)
9. Iterative Postorder Traversal | Set 2 (Using One Stack)
10. **Reverse Level Order Traversal**
11. Print Postorder traversal from given Inorder and Preorder traversals
12. **Level order traversal line by line | Set 2 (Using Two Queues)**
13. **Diagonal Traversal of Binary Tree**
14. Inorder Non-threaded Binary Tree Traversal without Recursion or Stack
15. **Check if leaf traversal of two Binary Trees is same?**
16. **Print a Binary Tree in Vertical Order | Set 1**
17. **Print a Binary Tree in Vertical Order | Set 2 (Hashmap based Method)**
18. Boundary Traversal of binary tree
19. Perfect Binary Tree Specific Level Order Traversal
20. Perfect Binary Tree Specific Level Order Traversal | Set 2
21. **If you are given two traversal sequences, can you construct the binary tree?**

Construction & Conversion:

1. Construct Tree from given Inorder and Preorder traversals
2. Construct a tree from Inorder and Level order traversals
3. **Construct Complete Binary Tree from its Linked List Representation**
4. Construct Full Binary Tree from given preorder and postorder traversals
5. Construct a special tree from given preorder traversal
6. Construct tree from ancestor matrix
7. Construct Ancestor Matrix from a Given Binary Tree
8. Construct Special Binary Tree from given Inorder traversal
9. Construct Binary Tree from given Parent Array representation
10. Construct a Binary Tree from Postorder and Inorder
11. Create a Doubly Linked List from a Ternary Tree
12. Creating a tree with Left-Child Right-Sibling Representation
13. Convert a given Binary Tree to Doubly Linked List | Set 1
14. Convert a given Binary Tree to Doubly Linked List | Set 2
15. Convert a given Binary Tree to Doubly Linked List | Set 3
16. Convert a given Binary Tree to Doubly Linked List | Set 4
17. Convert an arbitrary Binary Tree to a tree that holds Children Sum Property
18. Convert a Binary Tree to Threaded binary tree | Set 1 (Using Queue)
19. Convert a Binary Tree to Threaded binary tree | Set 2 (Efficient)
20. Convert left-right representation of a binary tree to down-right
21. **Convert a given tree to its Sum Tree**
22. **Change a Binary Tree so that every node stores sum of all nodes in left subtree**
23. **Write an Efficient Function to Convert a Binary Tree into its Mirror Tree**
24. Convert a normal BST to Balanced BST
25. Convert a Binary Tree into Doubly Linked List in spiral fashion
26. Convert a Binary Tree to a Circular Doubly Link List
27. Convert a tree to forest of even nodes
28. **Convert a given Binary tree to a tree that holds Logical AND property**
29. Convert Ternary Expression to a Binary Tree

**Basic :**

1. **Binary Search Tree | Set 1 (Search and Insertion)**
2. **Binary Search Tree | Set 2 (Delete)**
3. Advantages of BST over Hash Table

**Construction and Conversion :**

1. Construct BST from given preorder traversal | Set 1
2. Construct BST from given preorder traversal | Set 2
3. **Binary Tree to Binary Search Tree Conversion**
4. **Convert a BST to a Binary Tree such that sum of all greater keys is added to every key**
5. Sorted Linked List to Balanced BST
6. **Sorted Array to Balanced BST**
7. **Transform a BST to greater sum tree**
8. Construct all possible BSTs for keys 1 to N
9. **Convert a BST to a Binary Tree such that sum of all greater keys is added to every key**
10. In-place Convert BST into a Min-Heap
11. **Convert BST to Min Heap**
12. Construct BST from its given level order traversal **Check and Smallest/Largest Element :**
13. **A program to check if a binary tree is BST or not**
14. **Find k-th smallest element in BST (Order Statistics in BST)**
15. Check if each internal node of a BST has exactly one child
16. Check for Identical BSTs without building the trees
17. **K’th Largest Element in BST when modification to BST is not allowed**
18. **Second largest element in BST**
19. **K’th smallest element in BST using O(1) Extra Space**
20. **Check if given sorted sub-sequence exists in binary search tree**
21. Check whether BST contains Dead End or not
22. **Check if an array represents Inorder of Binary Search tree or not**
23. **Check if two BSTs contain same set of elements**
24. Largest number in BST which is less than or equal to N  
      
    **Red Black Tree and Threaded Binary Tree**  
    C Program for Red Black Tree Insertion  
    Left Leaning Red Black Tree (Insertion)  
    **Threaded Binary Tree**Threaded Binary Tree | Insertion

Threaded Binary Search Tree | Deletion

**Misc :**

1. **Find the node with minimum value in a Binary Search Tree**
2. Total number of possible Binary Search Trees with n keys
3. Sorted order printing of a given array that represents a BST
4. Inorder Successor in Binary Search Tree
5. **Print BST keys in the given range**
6. Find the largest BST subtree in a given Binary Tree | Set 1
7. Merge Two Balanced Binary Search Trees
8. Merge two BSTs with limited extra space
9. Two nodes of a BST are swapped, correct the BST
10. Floor and Ceil from a BST
11. Find if there is a triplet in a Balanced BST that adds to zero
12. Find a pair with given sum in a Balanced BST
13. Remove BST keys outside the given range
14. **Add all greater values to every node in a given BST**
15. **Inorder predecessor and successor for a given key in BST**
16. Given n appointments, find all conflicting appointments
17. **How to handle duplicates in Binary Search Tree?**
18. Data Structure for a single resource reservations
19. **Count BST nodes that lie in a given range**
20. Count BST subtrees that lie in given range
21. **How to implement decrease key or change key in Binary Search Tree?**
22. Print Common Nodes in Two Binary Search Trees
23. Count inversions in an array | Set 2 (Using Self-Balancing BST)
24. Replace every element with the least greater element on its right
25. Find pairs with given sum such that pair elements lie in different BSTs
26. Find the closest element in Binary Search Tree
27. **Sum of k smallest elements in BST**
28. **Maximum element between two nodes of BST**
29. Binary Search Tree insert with Parent Pointer
30. Largest BST in a Binary Tree | Set 2
31. Leaf nodes from Preorder of a Binary Search Tree
32. Find median of BST in O(n) time and O(1) space
33. **Remove all leaf nodes from the binary search tree**
34. Count pairs from two BSTs whose sum is equal to a given value x
35. **Find distance between two nodes of a Binary Search Tree**
36. Minimum Possible value of |ai + aj – k| for given array and k.

**Stack Data Structure**

Design and Implementation :

1. **Implement Queue using Stacks**
2. **Design and Implement Special Stack Data Structure | Added Space Optimized Version**
3. **Implement two stacks in an array**
4. **Implement Stack using Queues**
5. **Design a stack with operations on middle element**
6. How to efficiently implement k stacks in a single array?
7. **How to create mergable stack?**
8. Design a stack that supports getMin() in O(1) time and O(1) extra space
9. **Implement a stack using single queue**
10. How to implement stack using priority queue or heap?
11. Create a customized data structure which evaluates functions in O(1)

**Standard Problems based on Stack**

* Infix to Postfix Conversion using Stack
* The Stock Span Problem
* **Check for balanced parentheses in an expression**
* Next Greater Element
* The Celebrity Problem
* Expression Evaluation
* **Evaluation of Postfix Expression**
* Iterative Tower of Hanoi

Operations on Stack

* **Reverse a stack using recursion**
* **Sort a stack using recursion**
* Sort a stack using a temporary stack
* **Reverse a stack without using extra space in O(n)**

Misc :

* Iterative Postorder Traversal | Set 1 (Using Two Stacks)
* Iterative Postorder Traversal | Set 2 (Using One Stack)
* Merge Overlapping Intervals
* Largest Rectangular Area in a Histogram | Set 2
* Print ancestors of a given binary tree node without recursion
* **Reverse a string using stack**
* Program for Tower of Hanoi
* **Find maximum depth of nested parenthesis in a string**
* Find maximum of minimum for every window size in a given array
* **Length of the longest valid substring**
* **Iterative Depth First Traversal of Graph**
* Minimum number of bracket reversals needed to make an expression balanced
* Check if a given array can represent Preorder Traversal of Binary Search Tree
* Form minimum number from given sequence
* **Find if an expression has duplicate parenthesis or not**
* Find maximum difference between nearest left and right smaller elements
* Find next Smaller of next Greater in an array
* **Find maximum sum possible equal sum of three stacks**
* Count natural numbers whose all permutation are greater than that number
* **Delete consecutive same words in a sequence**
* Decode a string recursively encoded as count followed by substring
* Bubble sort using two Stacks
* Pattern Occurrences : Stack Implementation Java
* Iterative method to find ancestors of a given binary tree
* Stack Permutations (Check if an array is stack permutation of other)
* **Tracking current Maximum Element in a Stack**
* Check mirror in n-ary tree
* **Reverse a number using stack**
* **Reversing the first K elements of a Queue**
* **Reversing a Queue**
* **Check if stack elements are pairwise consecutive**
* Spaghetti Stack
* **Interleave the first half of the queue with second half**
* Remove brackets from an algebraic string containing + and – operators

**Queue Data Structure**

**Introduction :**

1. **Queue**
2. **Applications of Queue Data Structure**
3. **Priority Queue**
4. **Applications of Priority Queue**
5. **Deque**
6. Circular Queue

**Implementation :**

1. **Implement Queue using Stacks**
2. **LRU Cache Implementation**
3. **Implement Stack using Queues**
4. **Queue | Set 2 (Linked List Implementation)**
5. How to efficiently implement k Queues in a single array?
6. **Implement a stack using single queue**
7. Implementation of Deque using circular array
8. **Circular Queue | Set 2 (Circular Linked List Implementation)**

**Standard Problems :**

1. **Breadth First Traversal or BFS for a Graph**
2. **Level Order Tree Traversal**
3. **Construct Complete Binary Tree from its Linked List Representation**
4. Program for Page Replacement Algorithms | Set 2 (FIFO)
5. **Check whether a given Binary Tree is Complete or not | Set 1 (Iterative Solution)**

**Operations on Queue :**

1. **Reversing a Queue**
2. **Reversing the first K elements of a Queue**
3. **Interleave the first half of the queue with second half**

**Misc :**

1. **Level order traversal in spiral form**
2. Sliding Window Maximum (Maximum of all subarrays of size k)
3. Find the largest multiple of 3 | Set 1 (Using Queue)
4. Find the first circular tour that visits all petrol pumps
5. **Iterative Method to find Height of Binary Tree**
6. Implement PriorityQueue through Comparator in Java
7. An Interesting Method to Generate Binary Numbers from 1 to n
8. Minimum time required to rot all oranges
9. **Find maximum level sum in Binary Tree**
10. Sum of minimum and maximum elements of all subarrays of size k.
11. Distance of nearest cell having 1 in a binary matrix
12. **Level order traversal line by line | Set 2 (Using Two Queues)**
13. First negative integer in every window of size k
14. Minimum sum of squares of character counts in a given string after removing k characters
15. Queue based approach for first non-repeating character in a stream
16. **Averages of Levels in Binary Tree**
17. Stack Permutations (Check if an array is stack permutation of other)
18. Check if all levels of two trees are anagrams or not
19. Check mirror in n-ary tree

**Singly Linked List :**

1. **Introduction to Linked List**
2. **Linked List vs Array**
3. **Linked List Insertion**
4. **Linked List Deletion (Deleting a given key)**
5. **Linked List Deletion (Deleting a key at given position)**
6. **Find Length of a Linked List (Iterative and Recursive)**
7. **Search an element in a Linked List (Iterative and Recursive)**
8. **Swap nodes in a linked list without swapping data**
9. **Write a function to get Nth node in a Linked List**
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. **Merge two sorted linked lists**
17. Generic Linked List in C
18. **Function to check if a singly linked list is palindrome**
19. **Intersection point of two Linked Lists.**
20. **Recursive function to print reverse of a Linked List**
21. **Remove duplicates from a sorted linked list**
22. **Remove duplicates from an unsorted linked list**
23. **Pairwise swap elements of a given linked list**
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. **Identical Linked Lists**
29. Merge Sort for Linked Lists
30. Reverse a Linked List in groups of given size
31. Reverse alternate K nodes in a Singly Linked List
32. **Delete nodes which have a greater value on right side**
33. **Segregate even and odd nodes in a Linked List**
34. Detect and Remove Loop in a Linked List
35. **Add two numbers represented by linked lists | Set 1**
36. **Delete a given node in Linked List under given constraints**
37. Union and Intersection of two Linked Lists
38. Find a triplet from three linked lists with sum equal to a given number
39. **Rotate a Linked List**
40. Flattening a Linked List
41. Add two numbers represented by linked lists | Set 2
42. **Sort a linked list of 0s, 1s and 2s**
43. Flatten a multilevel linked list
44. **Delete N nodes after M nodes of a linked list**
45. QuickSort on Singly Linked List
46. **Merge a linked list into another linked list at alternate positions**
47. **Pairwise swap elements of a given linked list by changing links**
48. Given a linked list of line segments, remove middle points
49. Clone a linked list with next and random pointer | Set 1
50. Clone a linked list with next and random pointer | Set 2
51. **Insertion Sort for Singly Linked List**
52. Point to next higher value node in a linked list with an arbitrary pointer
53. **Rearrange a given linked list in-place.**
54. **Sort a linked list that is sorted alternating ascending and descending orders.**
55. Select a Random Node from a Singly Linked List
56. **Merge two sorted linked lists such that merged list is in reverse order**
57. **Compare two strings represented as linked lists**
58. **Rearrange a linked list such that all even and odd positioned nodes are together**
59. Rearrange a Linked List in Zig-Zag fashion
60. **Add 1 to a number represented as linked list**
61. Point arbit pointer to greatest value right side node in a linked list
62. **Merge two sorted linked lists such that merged list is in reverse order**
63. **Check if a linked list of strings forms a palindrome**
64. **Sort linked list which is already sorted on absolute values**
65. **Delete last occurrence of an item from linked list**
66. **Delete a Linked List node at a given position**
67. **Linked List in java**
68. **In-place Merge two linked lists without changing links of first list**
69. **Delete middle of linked list**
70. Merge K sorted linked lists | Set 1
71. **Decimal Equivalent of Binary Linked List**
72. Flatten a multi-level linked list | Set 2 (Depth wise)
73. Rearrange a given list such that it consists of alternating minimum maximum elements
74. Subtract Two Numbers represented as Linked Lists
75. Find pair for given sum in a sorted singly linked without extra space
76. Iteratively Reverse a linked list using only 2 pointers (An Interesting Method)
77. Partitioning a linked list around a given value and keeping the original order
78. Check linked list with a loop is palindrome or not
79. Clone a linked list with next and random pointer in O(1) space
80. Length of longest palindrome list in a linked list using O(1) extra space
81. Adding two polynomials using Linked List
82. Implementing Iterator pattern of a single Linked List
83. **Move all occurrences of an element to end in a linked list**
84. **Remove all occurrences of duplicates from a sorted Linked List**
85. **Remove every k-th node of the linked list**
86. **Check whether the length of given linked list is Even or Odd**
87. Union and Intersection of two linked lists | Set-2 (Using Merge Sort)
88. Multiply two numbers represented by Linked Lists
89. Union and Intersection of two linked lists | Set-3 (Hashing)
90. Find the sum of last n nodes of the given Linked List
91. Count pairs from two linked lists whose sum is equal to a given value
92. Merge k sorted linked lists | Set 2 (Using Min Heap)
93. Recursive selection sort for singly linked list | Swapping node links
94. **Find length of loop in linked list**
95. Reverse a Linked List in groups of given size | Set 2
96. **Insert node into the middle of the linked list**
97. Merge two sorted lists (in-place)
98. **Sort a linked list of 0s, 1s and 2s by changing links**
99. **Insert a node after the n-th node from the end**
100. Rotate Linked List block wise
101. **Count rotations in sorted and rotated linked list**
102. **Make middle node head in a linked list**

Circular Linked List:

1. **Circular Linked List Introduction and Applications,**
2. **Circular Linked List Traversal**
3. **Split a Circular Linked List into two halves**
4. **Sorted insert for circular linked list**
5. **Check if a linked list is Circular Linked List**
6. Convert a Binary Tree to a Circular Doubly Link List
7. **Circular Singly Linked List | Insertion**
8. **Deletion from a Circular Linked List**
9. Circular Queue | Set 2 (Circular Linked List Implementation)
10. **Count nodes in Circular linked list**
11. Josephus Circle using circular linked list
12. **Convert singly linked list into circular linked list**

Doubly Linked List:

1. **Doubly Linked List Introduction and Insertion**
2. **Delete a node in a Doubly Linked List**
3. **Reverse a Doubly Linked List**
4. The Great Tree-List Recursion Problem.
5. Copy a linked list with next and arbit pointer
6. QuickSort on Doubly Linked List
7. Swap Kth node from beginning with Kth node from end in a Linked List
8. Merge Sort for Doubly Linked List
9. Create a Doubly Linked List from a Ternary Tree
10. **Find pairs with given sum in doubly linked list**
11. **Insert value in sorted way in a sorted doubly linked list**
12. **Delete a Doubly Linked List node at a given position**
13. Count triplets in a sorted doubly linked list whose sum is equal to a given value x
14. **Remove duplicates from a sorted doubly linked list**
15. **Delete all occurrences of a given key in a doubly linked list**
16. Remove duplicates from an unsorted doubly linked list
17. Sort the biotonic doubly linked list
18. Sort a k sorted doubly linked list
19. Convert a given Binary Tree to Doubly Linked List | Set

Misc :

1. Skip List | Set 1 (Introduction)
2. Skip List | Set 2 (Insertion)
3. Skip List | Set 3 (Searching and Deletion)
4. **Reverse a stack without using extra space in O(n)**
5. **An interesting method to print reverse of a linked list**
6. Linked List representation of Disjoint Set Data Structures
7. Sublist Search (Search a linked list in another list)
8. **Doubly Circular Linked List | Set 1 (Introduction and Insertion)**
9. **Doubly Circular Linked List | Set 2 (Deletion)**
10. **How to insert elements in C++ STL List ?**
11. Unrolled Linked List | Set 1 (Introduction)
12. **A Programmer’s approach of looking at Array vs. Linked List**
13. **How to write C functions that modify head pointer of a Linked List?**
14. **Given a linked list which is sorted, how will you insert in sorted way**
15. **Can we reverse a linked list in less than O(n)?**
16. **Practice questions for Linked List and Recursion**
17. Construct a Maximum Sum Linked List out of two Sorted Linked Lists having some Common nodes
18. **Given only a pointer to a node to be deleted in a singly linked list, how do you delete it?**
19. Why Quick Sort preferred for Arrays and Merge Sort for Linked Lists?
20. **Squareroot(n)-th node in a Linked List**
21. **Find the fractional (or n/k – th) node in linked list**
22. **Find modular node in a linked list**
23. Construct a linked list from 2D matrix
24. **Find smallest and largest elements in singly linked list**
25. **Arrange consonants and vowels nodes in a linked list**
26. **Partitioning a linked list around a given value and If we don’t care about making the elements of the list “stable”**
27. **Modify contents of Linked List**

**Bitwise Algorithms**

Basic :

1. Find the element that appears once
2. **Detect if two integers have opposite signs**
3. Add 1 to a given number
4. **Multiply a given Integer with 3.5**
5. **Turn off the rightmost set bit**
6. **Find whether a given number is a power of 4 or not**
7. **Compute modulus division by a power-of-2-number**
8. **Rotate bits of a number**
9. **Find the Number Occurring Odd Number of Times**
10. **Check for Integer Overflow**
11. **Little and Big Endian Mystery**
12. **Count set bits in an integer**
13. **Count number of bits to be flipped to convert A to B**
14. **Efficient way to multiply with 7**
15. **Program to find whether a no is power of two**
16. **Position of rightmost set bit**
17. **Binary representation of a given number**
18. **Find position of the only set bit**
19. **How to swap two numbers without using a temporary variable?**
20. **Swap two nibbles in a byte**
21. **How to turn off a particular bit in a number?**
22. Russian Peasant (Multiply two numbers using bitwise operators)
23. Add two bit strings
24. **Write your own strcmp that ignores cases**
25. **Check if two numbers are equal without using arithmetic and comparison operators**
26. **Find XOR of two number without using XOR operator**
27. **XOR counts of 0s and 1s in binary representation**
28. Calculate XOR from 1 to n
29. **Multiply a number with 10 without using multiplication operator**
30. Equal Sum and XOR
31. Swap three variables without using temporary variable
32. **Check if a number has bits in alternate pattern**
33. Count minimum bits to flip such that XOR of A and B equal to C
34. **Efficient method for 2’s complement of a binary string**
35. **Toggle case of a string using Bitwise operators**
36. **Toggling k-th bit of a number**
37. Convert decimal fraction to binary number
38. **Toggle all the bits of a number except k-th bit**
39. **Set the rightmost unset bit**
40. Convert a binary number to octal
41. **Check in binary array the number represented by a subarray is odd or even**
42. **Toggle the last m bits**
43. 1 to n bit numbers with no consecutive 1s in binary representation
44. **Toggle bits in the given range**
45. **Unset bits in the given range**
46. **Find the largest number with n set and m unset bits**
47. **Find the smallest number with n set and m unset bits**
48. Sum of numbers with exactly 2 bits set
49. **Check if binary representation of a given number and its complement are anagram**
50. Josephus Problem Using Bit Magic

Intermediate :

1. **Swap bits in a given number**
2. **Add two numbers without using arithmetic operators**
3. **Smallest of three integers without comparison operators**
4. A Boolean Array Puzzle
5. **Compute the integer absolute value (abs) without branching**
6. **Compute the minimum or maximum of two integers without branching**
7. **Find the two non-repeating elements in an array of repeating elements**
8. **Write an Efficient C Program to Reverse Bits of a Number**
9. **Smallest power of 2 greater than or equal to n**
10. Write an Efficient Method to Check if a Number is Multiple of 3
11. **Write a C program to find the parity of an unsigned integer**
12. **Swap all odd and even bits**
13. Check if a number is multiple of 9 using bitwise operators
14. **Check if binary representation of a number is palindrome**
15. Generate n-bit Gray Codes
16. **Check if a given number is sparse or not**
17. Euclid’s Algorithm when % and / operations are costly
18. Find nth Magic Number
19. **How to swap two bits in a given integer?**
20. **Calculate 7n/8 without using division and multiplication operators**
21. Calculate square of a number without using \*, / and pow()
22. Generate 0 and 1 with 25% and 75% probability
23. **Find even occurring elements in an array of limited range**
24. Cyclic Redundancy Check and Modulo-2 Division
25. **Copy set bits in a range**
26. Check if a number is Bleak
27. Count strings with consecutive 1’s
28. Gray to Binary and Binary to Gray conversion
29. Find Next Sparse Number
30. Sum of bit differences among all pairs
31. Sum of Bitwise And of all pairs in a given array
32. Bitwise and (or &) of a range
33. Multiples of 4 (An Interesting Method)
34. Length of the Longest Consecutive 1s in Binary Representation
35. Pairs of complete strings in two sets of strings
36. Find profession in a special family
37. Print first n numbers with exactly two set bits
38. Check if bits of a number has count of consecutive set bits in increasing order
39. Subset sum queries using bitset
40. Maximum 0’s between two immediate 1’s in binary representation
41. Count all pairs of an array which differ in K bits
42. **Efficiently check if a string has duplicates without using any additional data structure**
43. Count trailing zero bits using lookup table
44. Count smaller numbers whose XOR with n produces greater value
45. Check divisibility in a binary stream
46. **Multiplication of two numbers with shift operator**
47. Determine if a string has all Unique Characters
48. Reverse an array without using subtract sign ‘-‘ anywhere in the code
49. Count numbers whose sum with x is equal to XOR with x
50. Maximum XOR value of a pair from a range
51. Numbers whose bitwise OR and sum with N are equal
52. Change bits to make specific OR value
53. Count smaller values whose XOR with x is greater than x
54. **Next greater integer having one more number of set bits**
55. Check if two numbers are bit rotations of each other or not
56. Previous smaller integer having one less number of set bits
57. **Check if binary representations of two numbers are anagram**
58. **Maximize a given unsigned number number by swapping bits at it’s extreme positions**
59. **Set bits in N equals to M in the given range**

Hard :

1. Count total set bits in all numbers from 1 to n
2. Program to count number of set bits in an (big) array
3. Next higher number with same number of set bits
4. Karatsuba algorithm for fast multiplication
5. Find the maximum subarray XOR in a given array
6. Inserting m into n such that m starts at bit j and ends at bit i
7. Find Duplicates of array using bit array
8. Find longest sequence of 1’s in binary representation with one flip
9. Closest (or Next) smaller and greater numbers with same number of set bits
10. Bitmasking and Dynamic Programming | Set-2 (TSP)
11. Compute the parity of a number using XOR and table look-up

Misc :

1. Interesting Facts about Bitwise Operators in C
2. Optimization Techniques | Set 1 (Modulus)
3. What are the differences between bitwise and logical AND operators in C/C++?
4. Bit Fields in C
5. C++ bitset and its application
6. C++ bitset interesting facts
7. Builtin functions of GCC compiler
8. Bit Tricks for Competitive Programming

**Heap Data Structure**

1. **Binary Heap**
2. Time Complexity of building a heap
3. Applications of Heap Data Structure
4. Why is Binary Heap Preferred over BST for Priority Queue?
5. Binomial Heap
6. Fibonacci Heap
7. Leftist Heap
8. Heap Sort
9. K’th Largest Element in an array
10. Sort an almost sorted array/
11. Tournament Tree (Winner Tree) and Binary Heap
12. Check if a given Binary Tree is Heap
13. How to check if a given array represents a Binary Heap?
14. Print all elements in sorted order from row and column wise sorted matrix
15. Connect n ropes with minimum cost
16. Design an efficient data structure for given operations
17. Merge k sorted arrays | Set 1
18. Sort numbers stored on different machines

**Graph Data Structure And Algorithms**

1. **Graph and its representations**
2. **Breadth First Traversal for a Graph**
3. **Depth First Traversal for a Graph**
4. **Applications of Depth First Search**
5. **Applications of Breadth First Traversal**
6. Longest Path in a Directed Acyclic Graph
7. Find Mother Vertex in a Graph
8. Transitive Closure of a Graph using DFS
9. Find K cores of an undirected Graph
10. **Iterative Depth First Search**
11. Iterative Deepening Search(IDS) or Iterative Deepening Depth First Search(IDDFS)

Graph Cycle:

1. Detect Cycle in a Directed Graph
2. **Detect Cycle in a an Undirected Graph**
3. Detect cycle in an undirected graph
4. Detect cycle in a direct graph using colors
5. Assign directions to edges so that the directed graph remains acyclic
6. Detect a negative cycle in a Graph | (Bellman Ford)

Topological Sorting:

1. **Topological Sorting**
2. All topological sorts of a Directed Acyclic Graph
3. **Kahn’s Algorithm for Topological Sorting**

**Minimum Spanning Tree:**

1. **Prim’s Minimum Spanning Tree (MST))**
2. Applications of Minimum Spanning Tree Problem
3. **Prim’s MST for Adjacency List Representation**
4. **Kruskal’s Minimum Spanning Tree Algorithm**
5. Boruvka’s algorithm for Minimum Spanning Tree
6. Steiner Tree

Shortest Paths:

1. **Dijkstra’s shortest path algorithm**
2. Dijkstra’s Algorithm for Adjacency List Representation
3. Bellman–Ford Algorithm
4. Floyd Warshall Algorithm
5. Johnson’s algorithm for All-pairs shortest paths
6. Shortest Path in Directed Acyclic Graph
7. Some interesting shortest path questions,
8. Shortest path with exactly k edges in a directed and weighted graph
9. Dial’s Algorithm
10. Printing paths in Dijsktra’s Algorithm
11. Shortest path of a weighted graph where weight is 1 or 2

**Connectivity:**

1. **Find if there is a path between two vertices in a directed graph**
2. Connectivity in a directed graph
3. Articulation Points (or Cut Vertices) in a Graph
4. Biconnected graph
5. Bridges in a graph
6. Eulerian path and circuit
7. Fleury’s Algorithm for printing Eulerian Path or Circuit
8. Strongly Connected Components
9. Transitive closure of a graph
10. Find the number of islands
11. Count all possible walks from a source to a destination with exactly k edges
12. Euler Circuit in a Directed Graph
13. Biconnected Components
14. Check if a given graph is tree or not
15. Karger’s algorithm for Minimum Cut
16. Find if there is a path of more than k length
17. Length of shortest chain to reach the target word
18. Print all paths from a given source to destination
19. Find minimum cost to reach destination using train
20. Tarjan’s Algorithm to find strongly connected Components

**Matrix:**

1. Search in a row wise and column wise sorted matrix
2. Print a given matrix in spiral form
3. A Boolean Matrix Question
4. Print unique rows in a given boolean matrix
5. Maximum size square sub-matrix with all 1s
6. Inplace M x N size matrix transpose | Updated
7. Print Matrix Diagonally
8. Dynamic Programming | Set 27 (Maximum sum rectangle in a 2D matrix)
9. Shift matrix elements row-wise by k
10. Strassen’s Matrix Multiplication
11. Create a matrix with alternating rectangles of O and X
12. Find the row with maximum number of 1s
13. Print all elements in sorted order from row and column wise sorted matrix
14. Given an n x n square matrix, find sum of all sub-squares of size k x k
15. Count number of islands where every island is row-wise and column-wise separated
16. Given a matrix of ‘O’ and ‘X’, replace ‘O’ with ‘X’ if surrounded by ‘X’
17. Find the longest path in a matrix with given constraints
18. Given a Boolean Matrix, find k such that all elements in k’th row are 0 and k’th column are 1.
19. Find the largest rectangle of 1’s with swapping of columns allowed
20. Validity of a given Tic-Tac-Toe board configuration
21. Minimum Initial Points to Reach Destination
22. Find length of the longest consecutive path from a given starting character
23. Collect maximum points in a grid using two traversals
24. Rotate Matrix Elements
25. Find sum of all elements in a matrix except the elements in row and/or column of given cell?
26. Find a common element in all rows of a given row-wise sorted matrix
27. Number of paths with exactly k coins
28. Collect maximum coins before hitting a dead end
29. Program for Rank of Matrix
30. Submatrix Sum Queries
31. Maximum size rectangle binary sub-matrix with all 1s
32. Count Negative Numbers in a Column-Wise and Row-Wise Sorted Matrix
33. Construct Ancestor Matrix from a Given Binary Tree
34. Construct tree from ancestor matrix
35. In-place convert matrix in specific order
36. Common elements in all rows of a given matrix
37. Print maximum sum square sub-matrix of given size
38. Find a specific pair in Matrix
39. Find orientation of a pattern in a matrix
40. Shortest path in a Binary Maze
41. Inplace rotate square matrix by 90 degrees
42. Return previous element in an expanding matrix

**String Data Structure**

1. **Strings in C**
2. **Storage for Strings in C**
3. String Class in C++
4. String Class in Java
5. **Function to copy string**
6. Strings in Python : Set 1, Set 2
7. String Methods in Python : Set 1 , Set 2 , Set 3
8. **Print reverse of a string using recursion**
9. **Reverse an array without affecting special characters**
10. **Remove all duplicates from a given string**
11. **Pangram Checking**
12. How to split a string in C/C++, Python and Java?
13. **Remove spaces from a given string**

**Character Counting Based Problems :**

1. **Return maximum occurring character in an input string**
2. Find the smallest window in a string containing all characters of another string
3. **C program to find second most frequent character**
4. K’th Non-repeating Character
5. Count number of substrings with exactly k distinct characters
6. Find k’th character of decrypted string
7. **Count characters at same position as in English alphabets**
8. Check if two strings are k-anagrams or not
9. **Count words in a given string**
10. Count of words whose i-th letter is either (i-1)-th, i-th, or (i+1)-th letter of given word
11. Count substrings with same first and last characters
12. **Maximum consecutive repeating character in string**
13. Count of strings that can be formed using a, b and c under given constraints
14. Group words with same set of characters
15. Count of total anagram substrings
16. Substring with highest frequency length product
17. Number of even substrings in a string of digits
18. **Print all distinct characters of a string in order (3 Methods)**
19. Smallest window that contains all characters of string itself
20. **Print common characters of two Strings in alphabetical order**
21. **Program to count occurrence of a given character in a string**
22. Minimum sum of squares of character counts in a given string after removing k characters
23. **Program to count vowels in a string (Iterative and Recursive)**
24. Number of distinct permutation a String can have
25. **Check if both halves of the string have same set of characters**
26. Count words that appear exactly two times in an array of words
27. Check if frequency of all characters can become same by one removal
28. Count ways to increase LCS length of two strings by one
29. **Print the string after the specified character has occurred given no. of times**
30. **Remove characters from the first string which are present in the second string**

**Anagram :**

1. **Check whether two strings are anagram of each other**
2. Given a sequence of words, print all anagrams together | Set 1
3. Given a sequence of words, print all anagrams together | Set 2
4. Anagram Substring Search
5. Print all pairs of anagrams in a given array of strings
6. **Remove minimum number of characters so that two strings become anagram**
7. **Check if two strings are k-anagrams or not**
8. **Check if binary representations of two numbers are anagram**
9. Given a sequence of words, print all anagrams together using STL
10. **Check if any anagram of a string is palindrome or not**
11. **Check if binary representation of a given number and its complement are anagram**
12. Covert string X to an anagram of string Y with minimum replacements
13. Count subsequences in first string which are anagrams of the second string
14. **Longest Common Anagram Subsequence**
15. Find the size of largest subset of anagram words
16. Check if all levels of two trees are anagrams or not
17. Anagram Substring Search (Or Search for all permutations)

**Palindrome :**

1. **C Program to Check if a Given String is Palindrome**
2. Check if a given string is a rotation of a palindrome
3. **C++ Program to print all palindromes in a given range**
4. **Check if characters of a given string can be rearranged to form a palindrome**
5. Dynamic Programming | Set 28 (Minimum insertions to form a palindrome)
6. Longest Palindromic Substring | Set 2
7. Print all palindromic partitions of a string
8. Find all distinct palindromic sub-strings of a given string
9. Online algorithm for checking palindrome in a stream
10. Given a string, print all possible palindromic partitions
11. Count All Palindromic Subsequence in a given String
12. Minimum characters to be added at front to make string palindrome
13. Palindrome Substring Queries
14. Suffix Tree Application 6 – Longest Palindromic Substring
15. Palindrome pair in an array of words (or strings)
16. Make largest palindrome by changing at most K-digits
17. Lexicographically first palindromic string
18. **Recursive function to check if a string is palindrome**
19. Minimum number of Appends needed to make a string palindrome
20. Longest Non-palindromic substring
21. Minimum number of deletions to make a string palindrome
22. Minimum steps to delete a string after repeated deletion of palindrome substrings
23. Count of Palindromic substrings in an Index range
24. Minimum insertions to form a palindrome with permutations allowed
25. Nth Even length Palindrome
26. Palindrome by swapping only one character
27. Find all distinct palindromic sub-strings of a given string
28. Print all funny words in a string
29. Print longest palindrome word in a sentence
30. Longest Palindromic Substring using Palindromic Tree | Set 3
31. Count palindromic characteristics of a String
32. Count palindrome words in a sentence
33. Largest palindromic number by permuting digits
34. Construct binary palindrome by repeated appending and trimming
35. Number of palindromic permutations | Set 1
36. Print the arranged positions of characters to make palindrome
37. Number of strings of length N with no palindromic sub string
38. Count special palindromes in a String
39. Closest Palindrome Number (absolute difference Is min)
40. Minimum removal to make palindrome permutation
41. Remove all the palindromic words from the given sentence
42. Rearrange characters to form palindrome if possible
43. Convert the string into palindrome string by changing only one character.
44. Count maximum-length palindromes in a String
45. Make a lexicographically smallest palindrome with minimal changes
46. Longest palindrome subsequence with O(n) space
47. Bash program to check if the Number is a Palindrome
48. Palindromic Selfie Numbers
49. Check if actual binary representation of a number is palindrome
50. Print all the palindromic permutations of given string in alphabetic order
51. Minimum equal palindromic cuts with rearrangements allowed
52. Form the largest palindromic number using atmost two swaps
53. Next word that does not contain a palindrome and has characters from first k
54. Number of palindromic subsequences of length k
55. Sum of first K even-length Palindrome numbers
56. Construct lexicographically smallest palindrome
57. Binary String of given length that without a palindrome of size 3
58. Minimum cost to convert string into palindrome
59. Double Base Palindrome
60. Minimum reduce operations to covert a given string into a palindrome
61. Check if number is palindrome or not in Octal4
62. Number of positions where a letter can be inserted such that a string becomes palindrome
63. Next higher palindromic number using the same set of digits
64. Print all palindromic paths from top left to bottom right in a matrix
65. Check if any anagram of a string is palindrome or not
66. To check a number is palindrome or not without using any extra space
67. Given two strings check which string makes a palindrome first
68. Sentence Palindrome (Palindrome after removing spaces, dots, .. etc)
69. Longest Possible Chunked Palindrome
70. Nth Even length Palindrome
71. Minimum insertions to form a palindrome with permutations allowed
72. OLA Interview Experience | Set 11 ( For Internship)
73. Count of Palindromic substrings in an Index range
74. Minimum steps to delete a string after repeated deletion of palindrome substrings
75. Minimum number of deletions to make a string palindrome
76. Longest Non-palindromic substring
77. Minimum number of Appends needed to make a string palindrome
78. Zoho Interview Experience | Set 21 (On-Campus)
79. Palindromic Tree | Introduction & Implementation
80. Lexicographically first palindromic string
81. N’th palindrome of K digits
82. Make largest palindrome by changing at most K-digits
83. Palindrome pair in an array of words (or strings)
84. Check if a doubly linked list of characters is palindrome or not
85. Length of longest palindrome list in a linked list using O(1) extra space
86. Check linked list with a loop is palindrome or not
87. Print all palindromic partitions of a string
88. Number of palindromic paths in a matrix
89. Generate all palindromic numbers less than n
90. Minimum characters to be added at front to make string palindrome
91. Count All Palindromic Subsequence in a given String
92. Palindromic Primes
93. Palindrome Substring Queries
94. Minimum number of palindromic subsequences to be removed to empty a binary string
95. Check if a linked list of strings forms a palindrome
96. Print all palindrome permutations of a string
97. Given a string, print all possible palindromic partitions
98. MakeMyTrip Interview Experience | Set 7 (On-Campus)
99. Online algorithm for checking palindrome in a stream
100. Find all distinct palindromic sub-strings of a given string
101. Suffix Tree Application 6 – Longest Palindromic Substring
102. Manacher’s Algorithm – Linear Time Longest Palindromic Substring – Part 4
103. Manacher’s Algorithm – Linear Time Longest Palindromic Substring – Part 3
104. Manacher’s Algorithm – Linear Time Longest Palindromic Substring – Part 2
105. Manacher’s Algorithm – Linear Time Longest Palindromic Substring – Part 1
106. Check if characters of a given string can be rearranged to form a palindrome
107. C++ Program to print all palindromes in a given range
108. Check if a given string is a rotation of a palindrome
109. C Program to Check if a Given String is Palindrome
110. Dynamic Programming | Set 28 (Minimum insertions to form a palindrome)
111. Facebook Interview | Set 1
112. Longest Palindromic Substring | Set 2
113. Given a number, find the next smallest palindrome
114. Dynamic Programming | Set 17 (Palindrome Partitioning)
115. Dynamic Programming | Set 12 (Longest Palindromic Subsequence)
116. Function to check if a singly linked list is palindrome

**Binary String :**

1. **Change if all bits can be made same by single flip**
2. Length of Longest sub-string that can be removed
3. Number of flips to make binary string alternate
4. **Efficient method for 2’s complement of a binary string**
5. Count binary strings with k times appearing adjacent two set bits
6. Given a binary string, count number of substrings that start and end with 1
7. Count strings with consecutive 1’s
8. Generate all binary strings from given pattern
9. Add two bit strings
10. Count number of binary strings without consecutive 1’s
11. Generate all binary permutations such that there are more or equal 1’s than 0’s before every point in all permutations
12. Check if a string follows a^nb^n pattern or not
13. **Binary representation of next number**
14. Binary representation of next greater number with same number of 1’s and 0’s
15. Decimal representation of given binary string is divisible by 5 or not
16. Check if a binary string has a 0 between 1s or not | Set 1
17. Check if a binary string has a 0 between 1s or not | Set 2
18. Min flips of continuous characters to make all characters same in a string
19. Program to add two binary strings
20. **Convert String into Binary Sequence**
21. Ways to remove one element from a binary string so that XOR becomes zero
22. Minimum bit changes in Binary Circular array to reach a index
23. Sorting array with conditional swapping
24. Count binary strings with twice zeros in first half
25. Minimum rooms for m events of n batches with given schedule
26. Check divisibility of binary string by 2^k
27. Find nth term of the Dragon Curve Sequence
28. Thue-Morse sequence
29. Construct binary palindrome by repeated appending and trimming
30. Minimum swaps required to Sort Binary array
31. Sort 1 to N by swapping adjacent elements
32. Find i’th Index character in a binary string obtained after n iterations
33. Maximum difference of zeros and ones in binary string | Set 2 (O(n) time)
34. Counting even decimal value substrings in a binary string
35. Number of substrings with odd decimal value in a binary string
36. Count subarrays with equal number of 1’s and 0’s
37. Longest subsegment of ‘1’s formed by changing at most k ‘0’s
38. Maximum difference of zeros and ones in binary string
39. Count of operations to make a binary string”ab” free
40. Minimum flips to make all 1s in left and 0s in right | Set 2
41. Longest subsequence having equal numbers of 0 and 1
42. Binary String of given length that without a palindrome of size 3
43. Longest subsequence of the form 0\*1\*0\* in a binary string
44. Longest subsequence with no 0 after 1
45. Divisibility by 64 with removal of bits allowed
46. Check if an encoding represents a unique binary string
47. Length of the longest substring with equal 1s and 0s
48. Rearrange a binary string as alternate x and y occurrences
49. Count of substrings of a binary string containing K ones
50. Print N-bit binary numbers having more 1’s than 0’s in all prefixes
51. Sort a binary array using one traversal
52. Binary representation of previous number
53. All possible binary numbers of length n with equal sum in both halves
54. Binary array after M range toggle operations
55. Longest common substring in binary representation of two numbers
56. Count of occurrences of a “1(0+)1” pattern in a string
57. Ways to remove one element from a binary string so that XOR becomes zero
58. Find the index of first 1 in a sorted array of 0’s and 1’s
59. Min flips of continuous characters to make all characters same in a string
60. Queries for decimal values of subarrays of a binary array
61. Check if a binary string has a 0 between 1s or not | Set 2 (Regular Expression Approach)
62. Check if a binary string has a 0 between 1s or not | Set 1 (General approach)
63. Find the transition point in a binary array
64. Count minimum right flips to set all values in an array
65. Decimal representation of given binary string is divisible by 5 or not
66. Fill array with 1’s using minimum iterations of filling neighbors
67. Binary representation of next greater number with same number of 1’s and 0’s
68. Binary representation of next number
69. Check if a string follows a^nb^n pattern or not
70. Minimum toggles to partition a binary array so that it has first 0s then 1s
71. Count passing car pairs
72. Maximum consecutive one’s (or zeros) in a binary array
73. Count binary strings with k times appearing adjacent two set bits
74. Efficient method for 2’s complement of a binary string
75. Number of flips to make binary string alternate
76. Length of Longest sub-string that can be removed
77. **Change if all bits can be made same by single flip**
78. Find all even length binary sequences with same sum of first and second half bits
79. Generate all binary strings from given pattern
80. Minimum number of palindromic subsequences to be removed to empty a binary string
81. Count strings with consecutive 1’s
82. Given a binary string, count number of substrings that start and end with 1.
83. Count 1’s in a sorted binary array
84. Generate all binary permutations such that there are more or equal 1’s than 0’s before every point in all permutations
85. Count number of binary strings without consecutive 1’s
86. Add two bit strings

**Pattern Searching :**

1. Searching for Patterns | Set 1 (Naive Pattern Searching)
2. Searching for Patterns | Set 2 (KMP Algorithm)
3. Searching for Patterns | Set 3 (Rabin-Karp Algorithm)
4. Searching for Patterns | Set 5 (Finite Automata)
5. Pattern Searching | Set 7 (Boyer Moore Algorithm – Bad Character Heuristic)
6. String matching where one string contains wildcard characters
7. Anagram Substring Search
8. Manacher’s Algorithm – Linear Time Longest Palindromic Substring – Part 1
9. Manacher’s Algorithm – Linear Time Longest Palindromic Substring – Part 2
10. Manacher’s Algorithm – Linear Time Longest Palindromic Substring – Part 3
11. Manacher’s Algorithm – Linear Time Longest Palindromic Substring – Part 4
12. Z algorithm
13. Search a Word in a 2D Grid of characters
14. Find all occurrences of a given word in a matrix
15. Maximum length prefix of one string that occurs as subsequence in another
16. Wildcard Pattern Matching
17. Replace all occurrences of string AB with C without using extra space
18. Aho-Corasick Algorithm
19. Find all the patterns of “1(0+)1” in a given string | SET 1
20. Find all the patterns of “1(0+)1” in a given string | SET 2
21. DFA for Strings not ending with “THE”
22. Check if a string is substring of another
23. Program to replace a word with asterisks in a sentence
24. Dynamic Programming | Wildcard Pattern Matching | Linear Time and Constant Space
25. Pattern Searching using C++ library
26. Longest prefix which is also suffix
27. Splitting a Numeric String
28. Count of number of given string in 2D character array
29. Find minimum shift for longest common prefix
30. Frequency of a substring in a string
31. Count of occurrences of a “1(0+)1” pattern in a string
32. Boyer Moore Algorithm | Good Suffix heuristic
33. is\_permutation() in C++ and its application for anagram search
34. Match Expression where a single special character in pattern can match one or more characters
35. Maximum length prefix of one string that occurs as subsequence in another
36. Replace all occurrences of string AB with C without using extra space
37. Wildcard Pattern Matching
38. Find all occurrences of a given word in a matrix
39. Aho-Corasick Algorithm for Pattern Searching
40. kasai’s Algorithm for Construction of LCP array from Suffix Array
41. Search a Word in a 2D Grid of characters
42. Z algorithm (Linear time pattern searching Algorithm)
43. Online algorithm for checking palindrome in a stream
44. Suffix Tree Application 6 – Longest Palindromic Substring
45. Manacher’s Algorithm – Linear Time Longest Palindromic Substring – Part 4
46. Manacher’s Algorithm – Linear Time Longest Palindromic Substring – Part 3
47. Manacher’s Algorithm – Linear Time Longest Palindromic Substring – Part 2
48. Manacher’s Algorithm – Linear Time Longest Palindromic Substring – Part 1
49. Suffix Tree Application 5 – Longest Common Substring
50. Generalized Suffix Tree 1
51. Suffix Tree Application 4 – Build Linear Time Suffix Array
52. Suffix Tree Application 3 – Longest Repeated Substring
53. Suffix Tree Application 2 – Searching All Patterns
54. Suffix Tree Application 1 – Substring Check
55. Ukkonen’s Suffix Tree Construction – Part 6
56. Ukkonen’s Suffix Tree Construction – Part 5
57. Ukkonen’s Suffix Tree Construction – Part 4
58. Ukkonen’s Suffix Tree Construction – Part 3
59. Ukkonen’s Suffix Tree Construction – Part 2
60. Ukkonen’s Suffix Tree Construction – Part 1
61. Pattern Searching using a Trie of all Suffixes
62. Anagram Substring Search (Or Search for all permutations)
63. Suffix Array | Set 2 (nLogn Algorithm)
64. Suffix Array | Set 1 (Introduction)
65. String matching where one string contains wildcard characters
66. Pattern Searching | Set 8 (Suffix Tree Introduction)
67. Pattern Searching | Set 7 (Boyer Moore Algorithm – Bad Character Heuristic)
68. Pattern Searching | Set 6 (Efficient Construction of Finite Automata)
69. Pattern Searching | Set 6 (Efficient Construction of Finite Automata)
70. Searching for Patterns | Set 4 (A Naive Pattern Searching Question)
71. Searching for Patterns | Set 3 (Rabin-Karp Algorithm)
72. Searching for Patterns | Set 2 (KMP Algorithm)
73. Searching for Patterns | Set 1 (Naive Pattern Searching)

**Searching Algorithms**

**Searching Algorithms :**

1. **Linear Search**
2. **Binary Search**
3. Jump Search
4. Interpolation Search
5. Exponential Search
6. Sublist Search (Search a linked list in another list)
7. Fibonacci Search
8. The Ubiquitous Binary Search
9. Recursive program to linearly search an element in a given array
10. Recursive function to do substring search
11. Unbounded Binary Search Example (Find the point where a monotonically increasing function becomes positive first time)

Comparisons :

1. Linear Search vs Binary Search
2. Interpolation search vs Binary search
3. Why is Binary Search preferred over Ternary Search?

Coding Problems :

1. **Find the Missing Number**
2. Search an element in a sorted and rotated array
3. **Median of two sorted arrays**
4. Two elements whose sum is closest to zero
5. **Find the smallest and second smallest element in an array**
6. Maximum and minimum of an array using minimum number of comparisons
7. k largest(or smallest) elements in an array | added Min Heap method
8. **Ceiling in a sorted array**
9. **Count number of occurrences (or frequency) in a sorted array**
10. Find the repeating and the missing | Added 3 new methods
11. **Find a Fixed Point in a given array**
12. Find the maximum element in an array which is first increasing and then decreasing
13. Find a pair with the given difference
14. Find the k most frequent words from a file
15. Median of two sorted arrays of different sizes
16. Find a peak element
17. Given an array of of size n and a number k, find all elements that appear more than n/k times
18. Find the minimum element in a sorted and rotated array
19. Kth smallest element in a row-wise and column-wise sorted 2D array | Set 1
20. Find k closest elements to a given value
21. Search in an almost sorted array
22. A Problem in Many Binary Search Implementations
23. Find the first repeating element in an array of integers
24. Find common elements in three sorted arrays
25. **Count 1’s in a sorted binary array**
26. Given a sorted array and a number x, find the pair in array whose sum is closest to x
27. Find the closest pair from two sorted arrays
28. K’th Smallest/Largest Element in Unsorted Array | Set 1
29. K’th Smallest/Largest Element in Unsorted Array | Set 2 (Expected Linear Time)
30. K’th Smallest/Largest Element in Unsorted Array | Set 3 (Worst Case Linear Time)
31. **Find position of an element in a sorted array of infinite numbers**
32. Given a sorted and rotated array, find if there is a pair with a given sum
33. **Find the largest pair sum in an unsorted array**
34. Find the nearest smaller numbers on left side in an array
35. K’th largest element in a stream
36. Find a pair with maximum product in array of Integers
37. Find the element that appears once in a sorted array
38. Find the odd appearing element in O(Log n) time
39. Find the largest three elements in an array
40. Search an element in an array where difference between adjacent elements is 1
41. Find three closest elements from given three sorted arrays
42. Find the element before which all the elements are smaller than it, and after which all are greater
43. Binary Search for Rational Numbers without using floating point arithmetic
44. **Floor in a Sorted Array**
45. **Third largest element in an array of distinct elements**
46. Second minimum element using minimum comparisons
47. Queries for greater than and not less than
48. Efficient search in an array where difference between adjacent is 1
49. Print all possible sums of consecutive numbers with sum N
50. Minimum time required to produce m items
51. Make all array elements equal with minimum cost
52. **Check if there exist two elements in an array whose sum is equal to the sum of rest of the array**
53. Check if reversing a sub array make the array sorted
54. Find all triplets with zero sum
55. **Search, insert and delete in an unsorted array**
56. **Search, insert and delete in a sorted array**
57. **Move all occurrences of an element to end in a linked list**
58. Search in an array of strings where non-empty strings are sorted
59. Smallest Difference Triplet from Three arrays
60. Best First Search (Informed Search)

======================================================================================================================================================

**Array Rotations :**

1. Program for array rotation
2. **Reversal algorithm for array rotation**
3. Block swap algorithm for array rotation
4. **Program to cyclically rotate an array by one**
5. Search an element in a sorted and rotated array
6. Given a sorted and rotated array, find if there is a pair with a given sum
7. Find maximum value of Sum( i\*arr[i]) with only rotations on given array allowed
8. Maximum sum of i\*arr[i] among all rotations of a given array
9. Find the Rotation Count in Rotated Sorted array
10. Quickly find multiple left rotations of an array
11. Find the minimum element in a sorted and rotated array
12. **Reversal algorithm for right rotation of an array**
13. Find a rotation with maximum hamming distance
14. Queries on Left and Right Circular shift on array
15. Print left rotation of array in O(n) time and O(1) space
16. Find element at given index after a number of rotations
17. Split the array and add the first part to the end