

# Bookmarks

## Bookmarks Bar

## DSA Sheet

## Core Subjects

[DSA Sheet](#) [Puzzles](#) [Aptitude](#) [OS Playlist](#) [DBMS Playlist](#) [CN Playlist](#) [SQL Leetcode](#) [OS Articles](#)  
[DBMS Articles](#) [CN Articles](#) [SQL Articles](#)

## Getting Started

[Java Programming Basics](#) [Basic Problems Practice](#)

## C++ Programming Basics

[C++ Basics](#) [Basic Pointers in C++](#) [Pointers in C++](#)

## STL C++

[STL Basics](#) [Templates in C++](#) [Generic Programming in C++](#) [Iterators in C++](#) [Inbuilt Algorithms in C++](#) [Comparator in C++](#) [Pair in C++](#) [Tuples in C++](#) [Containers in C++](#) [Strings in C++](#) [String Tokenizer in C++](#) [Hashing in C++](#) [Custom Hash Function](#) [Functors in C++](#) [Priority Queue in C++](#) [Custom Priority Queue](#) [Sets in C++](#) [Bitset in C++](#)

## Basic Java Problems

[Java Practice Problems](#) [Hello World - I](#) [Hello World - II](#) [Java Comments](#) [Java Datatypes](#) [Java Inputs](#) [Next vs Nextline](#) [Concatenate Strings](#) [Java If Else](#) [Check Vowel](#) [Change Case of String](#) [Check Even Odd](#) [Java Switch Case](#) [Java Loops](#) [Swap Two Integers](#)

## Loops Basics

[Print Increasing](#) [Print Decreasing](#) [Sum of Series](#) [Sum of AP Series](#) [Sum of GP Series](#) [Factorial Problem](#) [Power Function](#) [Check Power of 2](#) [Nth Fibonacci](#) [Nth Tribonacci](#) [Print A to Z](#) [Fizz Buzz](#)

## Digit Traversals

[Count Digits of Number](#) [Reverse Integer](#) [Double Reversal](#) [Sum & Product of Digits](#) [Self Dividing Numbers](#) [Armstrong Number](#) [Rotate Digits of Number](#) [Rotate Digits of Number](#) [Inverse of Number](#)

## Basic Math Problems

[Stones Remove Game](#) [Divisor Game](#) [Digital Root](#) [Bulb Switcher](#) [Tournament Matches](#) [Count Odd In Range](#) [Max Water Bottles](#) [Pythagorean Triplets](#) [Factorial Trailing 0s](#) [Check Ugly Number](#) [Nth Term of GP](#) [Days Between Dates](#) [Day of the Year](#) [Day of the Week](#) [Next Closest Time](#)

## Pattern Printing

[Pattern Printing](#) [Stars Pattern](#) [Lower Right Triangle](#) [Upper Left Triangle](#) [Lower Right Triangle](#) [Upper Right Triangle](#) [Diamond Pattern](#) [Hollow Diamond](#) [Diamond Border](#) [Backward Slash](#) [Forward Slash](#) [Cross of Stars](#) [Cross of Numbers](#) [Table of Number](#) [Pattern of Numbers - I](#) [Pattern of Numbers - II](#) [Pattern of Numbers - III](#) [Fibonacci Pattern](#) [Mirror Image Triangle](#) [Arrow Pattern](#) [Empty Hour Glass](#) [Swastik Pattern](#) [W Pattern](#)

## Time & Space Complexity

[Time & Space Complexity - I](#) [Time & Space Complexity - II](#) [Algorithm Analysis](#) [Time Complexity](#) [Master's Theorem](#) [Space Complexity](#) [Complexity Practice](#)

## Arrays & Strings

[Arrays and Strings Practice](#)

## Array & String Theory

[Array Data Structure](#) [Java Arrays](#) [Java 1D Arrays](#) [Array Utility Class](#) [String Data Structure](#) [Java Strings](#) [String Pool in Java](#) [String Immutability](#) [Java vs C++ Strings](#) [Java StringBuffer](#) [Java StringBuilder](#) [String Buffer vs Builder](#) [Multidimensional Array](#) [Jagged Arrays](#) [Java Collections](#) [Iterable Interface](#) [Java ArrayList](#) [Java Vector](#) [String Tokenizer](#) [Java](#)

## Reverse Array or String

[Reverse Array](#) [Left Rotate Array](#) [Right Rotate Array](#) [Reverse String](#) [Reverse String Range](#) [Reverse Words - I](#) [Reverse Words - II](#) [Reverse Vowels](#) [Palindromic String](#) [Palindromic Integer](#) [Palindrome After Deletion](#)

## String to Integer

[Integer to String](#) [String to Integer](#) [String to Integer](#) [Valid Number](#) [Excel Column No to Title](#) [Excel Column Title to No](#)

## Linear Traversal

[Linear Search](#) [Linear Search](#) [Maximum & Minimum](#) [Maximum & Minimum](#) [Smallest Range](#) [Second Largest](#) [Valid Mountain Array](#) [Is Array Sorted](#) [Rotated](#) [Distribute Candies](#) [Count Equal Pairs](#) [Range Addition](#) [Visiting All Points](#) [Subarray or Substrings](#) [Sum of Odd Subarrays](#) [Celebrity Problem - I](#) [Celebrity Problem - II](#) [Non Decreasing Array](#) [Target Modulo Sum Pair](#)

## Prefix Sum Array

[Running Sum](#) [Highest Altitude Point](#) [Positive Step Sum](#) [Leaders in Array](#) [Equilibrium Point](#) [Balanced Array](#) [Product Array Puzzle](#) [Max Closest Person](#) [Sum of Even Nos Queries](#) [Prefix](#) [Aligned Binary String](#) [Shifting Letters](#) [Polynomial Evaluation](#) [Polynomial Evaluation](#) [Prefix Sum](#) [Array Range Sum Query 1D](#) [Range Sum Query 2D](#) [Range Sum Query](#) [Difference Array](#) [Range Update Queries - I](#) [Range Update Queries - II](#)

## Matrix or 2D Array

## Basic Matrix Traversal

[2D Array or Matrix Theory](#) [Print Matrix Row Wise](#) [Print Matrix Column Wise](#) [Wave or Snake Traversal](#) [Reverse Wave Traversal](#) [Searching in Matrix](#) [Exit Point in Matrix](#) [Convert 1D Array to 2D](#) [Shift 2D Grid](#) [Reshape Matrix](#)

## Two Matrices

[Identical Matrices](#) [Add Matrices](#) [Multiply Matrices](#)

## Rotate Matrix

[Transpose Matrix](#) [Rotate 90D Clockwise](#) [Rotate 90D Anticlockwise](#) [Rotate 180D Clockwise](#) [Rotate 180D Anticlockwise](#) [Equal Matrix Rotations](#)

## Diagonal Traversal

[Diagonal Traversal - BL to TR](#) [Diagonal Traversal - Zig Zag](#) [Diagonal Traversal - TL to BR](#) [Matrix Diagonal Sum](#) [Toeplitz Matrix](#) [Check X Matrix](#) [Upper Lower Triangle](#)

## Rows & Columns

[Saddle Point in Matrix](#) [Set Matrix Zeros](#) [Valid Tic Tac Toe State](#) [Winner of Tic Tac Toe](#) [Equal Row Column Sum](#) [Rows Columns Valid Matrix](#) [Game of Life](#)

## Spiral Matrix Traversal

[Boundary Traversal](#) [Spiral Matrix Traversal](#) [Fill Matrix in Spiral Order](#) [Reverse Spiral Matrix](#) [Spiral Matrix - Starting Pt](#) [Shell Rotate Matrix](#) [Shell Rotate Matrix](#)

## Big Integers as Arrays

[Plus One to Array](#) [Plus K to Array](#) [Add Two Strings](#) [Subtract Two Arrays](#) [Subtract Two Arrays](#)

[Compare Two Strings](#) [Multiply Two Strings](#) [Large Factorial](#) [Multiply Polynomials](#) [Sparse Matrix](#)  
[Sparse Matrix](#) [Big Integers Java](#) [Big Integers Java](#)

## Majority Element

[Majority Element - I](#) [Majority Element - II](#) [Majority Element - III](#)

## Missing & Duplicate

[Missing Number - I](#) [Missing Number - II](#) [First Missing Positive](#) [Repeating Element - I](#) [Repeating Element - II](#) [Repeating Element - III](#) [Max Repeating Element](#) [Missing & Repeating](#)

## String Matching Problems

[Is Subsequence](#) [Largest Word in Dictionary](#) [Valid Anagram](#) [K Anagrams](#) [Anagram Substrings](#)  
[Anagram Mappings](#) [Group Anagrams](#) [Group Shifted Strings](#) [Group Shifted Strings](#) [Find & Replace Pattern](#) [Odd - Even Anagrams](#) [Word Subsets](#) [Isomorphic Strings - I](#) [Isomorphic Strings - II](#) [Word Pattern - I](#) [Word Pattern - II](#) [Camelcase Matching](#) [First Unique Character](#) [Remove Duplicate Chars](#) [First Repeated Character](#)

## More String Problems

[Run Length Encoding](#) [String Compression](#) [Compare Version Nos](#) [Zig Zag Conversion](#) [Validate IP Address](#) [Maximum Number Swap](#) [Largest Odd Substring](#) [Beauty of All Substrings](#) [Pretty JSON](#)

## Recursion & Backtracking

### Recursion Basics

[Recursion Theory](#) [Backtracking Theory](#) [Print Decreasing](#) [Print Increasing](#) [Print Increasing Decreasing](#) [Factorial](#) [Print ZigZag](#) [Tower of Hanoi](#) [Exponentiation](#) [Binary Exponentiation](#) [Print Array Left to Right](#) [Print Array Right to Left](#) [Check Is Array Sorted](#) [Largest Element in Array](#) [First & Last Occurrence](#) [Find All Occurrences](#)

## Subsets Problems

[Subsets](#) [Subarray vs Subset](#) [Iterative Method](#) [Get Subsets](#) [Print Subsets](#) [Unique Subsets](#)  
[Subsets of Given Size](#) [Letter Tile Possibilities](#) [Square Matchsticks](#) [Beautiful Arrangement](#)

## Maze Path Problems

[Print Stair Paths](#) [Get Stair Paths](#) [All Maze Paths](#) [Rat in Maze](#) [Rat in Maze Count](#) [Rat in Maze Paths](#) [Rat in Maze with Jumps](#) [Get Maze Path with Jumps](#) [Print Maze Path with Jumps](#) [Gold Mine Problem](#)

## P & C Problems

## Combinations

[Combinations](#) [Combinations - 1](#) [Combinations - 2](#) [Factor Combinations](#)

## Permutations

[Permutations](#) [Permutations - 1](#) [Permutations - 2](#) [Permutations - 3](#) [Distinct Permutations](#) [Distinct Permutations - 1](#) [Distinct Permutations - 2](#) [Palindromic Permutations](#)

## K Words Combination

[K Words - Combination - I](#) [K Words - Combination - II](#) [K Words - Combination - III](#) [K Words - Combination - IV](#)

## K Words Permutation

[K Words - Permutation - I](#) [K Words - Permutation - II](#) [K Words - Permutation - III](#) [K Words - Permutation - IV](#)

## 2D Perm & Comb

[Permutations - 2d as 2d - Box Choose](#) [Combinations - 2d as 2d - Box Choose](#) [Permutations - 2d as 2d - Queen Choose](#) [Combination - 2d as 2d - Box Choose](#) [Combination - 2d as 1d - Queen Choose](#)

## Keypad Problems

[Keypad Combinations](#) [Get Keypad Combinations](#) [Print Keypad Combinations](#) [Print Decode Ways](#) [Print Decode Ways](#) [Generalized Abbreviations](#) [Abbreviations - Using R&B](#) [Abbreviations - Using Bits](#) [Keypad Sequence](#) [Letter Case Permutations](#)

## N Queen Problem

[N Queen - Combinations](#) [N Queen - Permutations](#) [N Queens - I](#) [N Queens - II](#) [N Queen - Backtracking](#) [N Queen - Branch & Bound](#) [N Queen - Bit Manipulation](#) [N Knights](#) [N Knights Knight's Tour Problem](#)

## Coin Change Problem

[Coin Change - Combination - 1](#) [Coin Change - Combination - 2](#) [Coin Change - Combination - 3](#) [Coin Change - Permutation - 1](#) [Coin Change - Permutation - 2](#)

## Puzzles Problems

[Josephus Problem](#) [Word Search](#) [Crossword Puzzle - I](#) [Crossword Puzzle - II](#) [Valid Sudoku](#) [Sudoku Solver](#) [Sudoku Solver](#) [Verbal Arithmetic Puzzle](#) [Magnet Puzzle](#) [Smallest No in DI Pattern](#) [Max Score Words](#)

## More Recursion Problems

[Largest No in K Swaps](#) [Print Friend Pairings](#) [Stepping Numbers](#) [Generate Balanced Parenthesis](#) [Lexicographical Numbers](#) [Restore IP Addresses](#) [Count & Say](#) [Convert Integer to Roman](#) [Convert Roman to Integer](#) [Integer in English Words](#) [Expression Add Operators](#) [Additive Numbers](#) [Split into Fibonacci Sequence](#) [Paranthesis Addition Ways](#) [Kth Symbol in Grammar](#)

[Lucky Number](#)

## Number Theory

### Euclid's Algorithm

[GCD of 2 Numbers](#) [Euclid's Algorithm](#) [Euclid's Algorithm](#) [Euclid's Algorithm](#) [GCD of Array](#) [GCD of Big Integer](#) [GCD of Big Integer](#) [Required Rooms](#) [Deck of Cards](#) [Simplified Fractions](#) [Min Deletions Array Divisible](#) [Cutting Squares](#) [Largest Coprime Divisor](#) [Largest Coprime Divisor](#)

### Extended Euclid

[Extended Euclid Algorithm](#) [Extended Euclid Algorithm](#) [Extended Euclid Algorithm](#) [Linear Diophantine Eqn](#) [Linear Diophantine Eqn](#) [Linear Diophantine Eqn](#)

### Prime Numbers

[Prime Numbers Playlist](#) [Checking No is Prime](#) [Kth Factor of Number](#) [Three Divisors](#) [Four Divisors](#) [Count Factors](#) [Sum of Factors](#) [Product of Factors](#) [Number & Sum of Factors](#) [Number & Sum of Factors](#) [Count Common Divisors](#) [Count Common Divisors](#) [Perfect Number](#) [Count Prime Numbers](#) [Sieve of Eratosthenes](#) [Find Kth Prime](#) [Prime Arrangements](#) [Smallest Prime Factor](#) [Largest Prime Factor](#) [Primes in Range I](#) [Bitset Sieve](#) [Product of Primes](#) [Primes in Range II](#) [Segmented Sieve](#) [Prime Factorization](#) [Prime Factorization Queries](#) [Count Prime Factors](#) [Divisor Queries](#) [Sieve Variations](#)

### Modular Arithmetic

[Modular Arithmetic](#) [Modular Arithmeitc](#) [Modular Factorial](#) [Modular Exponentiation](#) [Modular Exponentiation - I](#) [Modular Exponentiation - II](#) [Fermat's Little Theorem](#) [Fermat's Little Theorem](#) [Modular Division](#) [Modular Multiplicative Inverse](#) [Modular Multiplicative Inverse](#) [Modular Multiplicative Inverse](#) [Modular Binomial Coeff](#) [Modular Permutation Coeff](#) [Modular Permutation Coeff](#) [Sum of All Subarrays](#) [Sum of All Subsets](#) [Sum of Subset Width](#)

### Inclusion Exclusion Principle



[Inclusion Exclusion Principle](#) [Inclusion Exclusion Principle](#) [Nth Magical Number](#) [Nth Ugly Number](#)

## Sorting Algorithms

[Sorting Algo Theory - I](#) [Sorting Algo Theory - II](#) [Sort Array](#) [Inbuilt Sorting](#)

### Basic Sorting Algorithms

[Bubble Sort](#) [Bubble Sort](#) [Bubble Sort - Recursive](#) [Bubble Sort - Count Swaps](#) [Insertion Sort](#) [Insertion Sort](#) [Insertion Sort - Recursive](#) [Selection Sort](#) [Selection Sort](#) [Selection Sort - Recursive](#) [Shell Sort](#) [Shell Sort](#) [Pancake Sorting](#)

## Merge Sorted Arrays

[Merge Two Sorted Arrays - I](#) [Merging Using Two Pointers](#) [Merge Two Sorted Arrays - II](#) [Merging using Insertion Sort](#) [Merging using Shell Sort](#) [Union of 2 Sorted Arrays](#) [Intersection of 2 Arrays - I](#) [Intersection of 2 Arrays - II](#) [Intersection of 3 Arrays](#) [Symmetric Difference](#) [Squares of Sorted Array](#)

## Partitioning Algorithm

[Partition Around Pivot](#) [Sort Binary Array](#) [Sort Colors 012](#) [Dual Pivot Partitioning](#) [Three Way Partitioning](#) [Move Zeroes To End](#) [Segregate w/o Relative Order](#) [Segregate with Relative Order](#) [Segregate with Relative Order](#) [Alternate w/o Relative Order](#) [Alternate with Relative Order](#)

## Rearrange Array

[Wiggle Sort - 1](#) [Wiggle Sort - 1](#) [Wiggle Sort - 2](#) [Wave Sort - Lexical Order](#) [Wave Sort - Min Max Form](#) [Inverse Permutation - I](#) [Inverse Permutation - II](#) [Reorder Array with Indexes](#) [Reorder Array with Indexes](#)

## Divide & Conquer Based

[Merge Sort](#) [Merge Sort](#) [Quick Sort](#) [Quick Sort](#) [Dual Pivot Quicksort](#) [3 Way Quick Sort](#) [Improving Quicksort](#)

## Counting Sort

[Counting Sort](#) [Radix Sort](#) [Bucket Sort](#) [Sort Array of Dates](#) [Minimum Time Distance](#) [H Index](#) [Unsorted Height Checker](#) [Relative Sort Array](#) [Max Consecutive Gap](#) [Frequency Sort](#) [Frequency Sort String](#) [Top K Frequent Elements](#) [Top K Frequent Words](#) [Top K Frequent in Stream](#)

## Inversion Count

[Inversion Count](#) [Inversions - Merge Sort](#) [Inversions - Policy Based DS](#) [Policy Based DS](#) [Min Adjacent Swaps to Sort](#) [Global & Local Inversions](#) [Reverse Pairs](#) [Count of Smaller After Self](#) [Count of Range Sum](#) [Count Binary Substrings](#)

## Binary Search

### Binary Search Basics

[Binary Search Theory](#) [Binary Search](#) [Transition Point](#) [First Bad Version](#) [Guess No Higher or Lower](#) [Arranging Coins](#) [Fixed Point in Sorted Array](#)

### Lower & Upper Bound

[Lower & Upper Bound](#) [Search Insert Position](#) [Ceil in Sorted Array](#) [Floor in Sorted Array](#) [First & Last Occurrence](#) [Count Occurrences](#) [Closest Element](#) [K Closest Elements](#) [Heaters](#) [Remove 2\\*Min >= Max](#)

## Root & Square

[Square Root - Integral](#) [Square Root - Decimal](#) [Square Root](#) [Valid Perfect Square](#) [Nth Root of Number](#)

## Rotated Sorted Array

[Pivot in Rotated Sorted - I](#) [Pivot in Rotated Sorted - II](#) [Rotation Count in Rotated Sorted](#) [Search in Rotated Sorted - I](#) [Search in Rotated Sorted - II](#) [Search in Nearly Sorted](#) [Search in Nearly Sorted](#)

## Mountain Array

[Peak in Bitonic Array](#) [Search in Bitonic Array](#) [Find Peak Element](#)

## Search in 2D Matrix

[Search 2D Matrix - I](#) [Search 2D Matrix - II](#) [Count 0 in Binary Matrix](#) [Max 1s Row in Binary Matrix](#) [Peak Element in Matrix](#)

## Median in 2 Arrays

[Median 2 Sorted Arrays - Same Size](#) [Median 2 Sorted Arrays - Dff Size](#) [Kth Element in 2 Sorted Arrays](#) [Median in Row Wise Sorted Matrix](#)

## Binary Search on Answer

[Book Allocation](#) [Painter's Partition](#) [Split Array Largest Sum](#) [Capacity to Ship Packages](#) [Min Days to Make Bouquets](#) [Koko Eating Banana](#) [Smallest Divisor Threshold](#) [Min Time to Complete Trips](#) [Min Speed to Arrive on Time](#) [Aggressive Cows](#) [Magnetic Force Balls](#) [Max Candies to Children](#) [Woodcutting Eko - SPOJ](#) [Minimize Max Products Store](#) [Minimize Farthest Gas Stations](#) [Roti Prata - SPOJ](#)

## More Binary Search Problems

[Single Element in Sorted](#) [Kth Missing Element](#) [H Index - Sorted](#) [Longest Subset Limited Sum](#) [Kth Smallest Pair Distance](#) [Random Pick with Weight](#) [Missing Element of AP](#) [Min Factorial Trailing 0s](#) [Count Factorials with k 0s](#) [Left Out Candies](#) [Min Coin Piles](#) [Longest Subarray Sum > 0](#) [Double Helix - SPOJ](#)

## Linked List

## Design Linked List

[Design Linked List](#) [Singly Linked List Theory](#) [Doubly Linked List Theory](#) [Circular Linked List Theory](#) [Searching - SLL](#) [Traversal - SLL](#)

## Remove List Nodes

[Remove All Occurences](#) [Remove Duplicates in Sorted List - I](#) [Remove Duplicates in Sorted List - II](#) [Delete Node Without Head](#) [Remove 0 Sum Nodes](#)

## Two Pointers in List

[Get Middle Node of List](#) [Delete Middle Node of List](#) [Get Kth Node from End](#) [Delete Kth Node from End](#) [Swap Kth List Nodes](#) [Intersection Node of 2 Lists](#) [Common Nodes in 2 Lists](#) [Common Nodes in 2 Sorted Lists](#) [Merge 2 Unsorted Lists](#)

## Floyd's Cycle Detection

[Check Circular List](#) [Linked List Cycle Detection](#) [Starting List Node in Cycle](#) [Length of Linked List Cycle](#) [Remove Cycle from List](#)

## Reverse Linked List

[Reverse Singly Linked List](#) [Reverse SLL in Range](#) [Reverse SLL in K Groups](#) [Swap List Nodes in Pairs](#) [Reverse Even List Nodes](#) [Reverse Doubly Linked List](#) [Palindrome Linked List](#)

## Merge Sort List

[Merge 2 Sorted Lists](#) [Merge K Sorted Lists](#) [Flatten Sorted Lists](#) [Sort Singly Linked List](#) [Merge Sort - SLL](#) [Merge Sort - DLL](#)

## Quick Sort List

[Partition Linked List](#) [Sort Binary Linked List](#) [Segregate Odd & Even Nodes](#) [Sort Colors in Linked List](#) [QuickSort - SLL](#) [Quick Sort - DLL](#) [Quicksort vs Mergesort](#) [Quicksort vs Mergesort](#)

## Reorder Linked List

[Fold of Linked List](#) [Unfold of Linked List](#) [Rotate Linked List](#) [Insertion Sort List](#) [Split List in K Parts](#) [Flatten Multilevel DLL](#) [Clone List with Random Pointers](#) [Absolute List Sorting](#)

## Big Integers as Lists

[Add 1 to Linked List](#) [Add Two Lists - I](#) [Add Two Lists - II](#) [Subtract Two Lists](#) [Multiply Two Lists](#) [Polynomial Addition](#) [Multiply Polynomials](#)

## Design Cache

[Design LRU Cache](#) [LRU Page Faults](#) [Design LFU Cache](#) [Design GetRandom - I](#) [Design GetRandom - II](#)

## Stack & Queue

### Stack & Queue Basics

[Stacks Theory](#) [Design Stack Using Array](#) [Design Stack Using List](#) [Design 2 Stacks in Array](#) [Design K Stacks in Array](#) [Design Stack Using Heap](#) [Queue Theory](#) [Design Queue Using Array](#) [Design Queue Using List](#) [Design Circular Queue](#) [Circular Queue Using Array](#) [Circular Queue Using List](#)

### Doubly Ended Queue

[Deque Theory](#) [Design Circular Deque](#) [Deque Using Circular Array](#) [Deque Using DLL](#) [Stack & Queue using Deque](#)

### Design Stack or Queue

[Design Stack using Queue](#) [Design Queue using Stack](#) [Design Minimum Stack](#) [Design Maximum Stack](#) [Design Max Frequency Stack](#) [Design Middle Queue](#) [Delete Middle in Stack](#) [Design Stack with Increments](#) [Validate Stack Sequences](#)

## **Rearrange Stack or Queue**

[Reverse Stack](#) [Reverse Queue](#) [Sort Stack](#) [Sort Queue](#) [Interleave Queue](#)

## **Parenthesis Matching**

[Redundant Braces](#) [Balanced Parenthesis - I](#) [Balanced Parenthesis - II](#) [Remove Outermost Paranthesis](#) [Longest Valid Parentheses](#) [Min Additions for Balanced Braces - I](#) [Min Additions for Balanced Braces - II](#) [Min Removals for Balanced Braces - I](#) [Min Removals for Balanced Braces - II](#) [Min Reversals for Balanced Braces](#) [Min Swaps for Balanced Braces](#) [Score of Parenthesis](#) [Reverse Substrings B/W Braces](#) [Decode String](#) [Number of Atoms](#) [Maximum Nesting Depth](#)

## **Expressions Evaluation**

### **Infix Expressions**

[Infix Evaluation](#) [Infix Evaluation](#) [Infix Conversions](#) [Infix To Prefix Conversion](#) [Infix To Postfix Conversion](#) [Basic Calculator - I](#) [Basic Calculator - II](#) [Basic Calculator - III](#)

### **Postfix Expressions**

[Postfix Expression](#) [Postfix Evaluation](#) [Postfix to Infix Conversion](#) [Postfix to Prefix Conversion](#)

### **Prefix Expressions**

[Prefix Expression](#) [Prefix Evaluation](#) [Prefix to Infix Conversion](#) [Prefix to Postfix Conversion](#)

## **Expression Tree**

[Exp Tree - Construction](#) [Exp Tree - Evaluation](#) [Expression Tree](#)

## Monotonic Stack

[Next Greater Element](#) [NGE - Circular Array](#) [NGE - Two Arrays](#) [NGE - Linked List](#) [Prices with Special Discount](#) [Daily Temperatures](#) [Online Stock Span](#) [Largest Area Histogram](#) [132 Pattern](#) [Sum of Subarray Ranges](#) [Sum of Subarray Minimums](#) [Sum of Total Wizard Strength](#) [Max Subarray Min Product](#) [Max of Sliding Window Min](#) [No of Valid Subarrays](#) [Visible People in Queue](#) [Count Submatrix with All 1s](#) [Maximal Rectangle](#) [Remove Duplicate Letters](#) [Remove K Digits](#) [Lexico Smallest Subset](#)

## More Stack Queue Problems

[Asteroid Collision](#) [Exclusive Time of Functions](#) [Number of Recent Calls](#) [Valid Word After Substitution](#) [1st Unique Char - Stream](#) [Decode String at Index](#) [Generate Binary Numbers](#) [Remove Adjacent Duplicates - I](#) [Remove Adjacent Duplicates - II](#) [Remove Adjacent Duplicates - III](#) [Card Rotation - I](#) [Card Rotation - II](#)

## Hashmap & Heaps

## Hashing Technique

[Hashing Theory](#) [Separate Chaining](#) [Open Addressing](#) [Double Hashing](#) [Rehashing](#) [Hashing Technique](#) [Design HashMap](#) [Design HashSet](#)

## Intersection & Union

[Common Elements](#) [Uncommon Elements](#) [Union of Unsorted Arrays](#) [Common Elements in 2 of 3](#) [Contains Duplicate - I](#) [Contains Duplicate - II](#) [Contains Duplicate - III](#) [First Repeating Element](#) [First Unique Element](#)

## Subarray Sum

## Target Subarray Sum

[0 Sum Subarray](#) [Largest 0 Sum Subarray](#) [Count 0 Sum Subarrays](#) [K Sum Subarray](#) [K Sum Subarray - Only Positives](#) [K Sum Subarray - With Negatives](#) [Count K Sum Subarrays](#) [Largest K Sum Subarray](#) [Min Moves Reduce X to 0](#) [Count K XOR Subarrays](#)

## Divisible Subarrays

[Largest Divisible Subarray](#) [Count Divisible Subarrays](#) [K Multiple Subarray Sum](#) [Count Bad Pairs - I](#) [Count Bad Pairs - II](#) [Count K Absolute Diff Pairs](#) [Count Equal Divisible Pairs](#) [Count Equal Distance Points](#)

## Binary Subarray Sum

[Largest Subarray Equal 01](#) [Count Subarrays Equal 01](#) [Longest Subarray Equal 012](#) [Count Subarrays Equal 012](#) [Largest Subarray More 1s](#) [Binary Subarray Target Sum](#)

## Target Sum Submatrices

[Count 0 Sum Submatrix](#) [Largest 0 Sum Submatrix](#) [Count K Sum Submatrix](#) [Largest Submatrix Equal 01](#) [Max Sum Square Submatrix](#)

## More Hashing Questions

[Count 0 XOR Pairs](#) [Degree of Array](#) [Longest Consecutive Sequence](#) [Tricky Sorting Cost](#) [Array Pairs Divisible by K](#) [Fraction to Recurring Decimal](#) [Rabbits in Forest](#) [Array of Doubled Pairs](#) [Powerful Integers](#) [Hand of Straights](#) [Rank Transform of Array](#) [Stream Disjoint Intervals](#) [Brick Wall](#) [Max Equal Frequency](#)

## Binary Heap

[Heap Theory](#) [Priority Queue Theory](#) [Design Priority Queue](#) [Design Priority Queue](#) [Build Min Heap](#) [Build Min Heap](#) [Comparable & Comparator](#) [Comparable & Comparator](#) [Heap Sort](#) [Heap Sort MCQ](#) [Does Array Represent Heap](#) [Is Binary Tree Heap](#) [Convert Min to Max Heap](#) [Convert BST to Min Heap](#) [Merge Two Max Heaps](#) [Priority Queue Using List](#) [Priority Queue Using Heap](#)



## Heap Order Statistics

[K Largest - Sorted Order](#) [K Largest - Original Order](#) [K Closest Points](#) [Kth Smallest Element](#) [Kth Largest Element](#) [Quick Select Algorithm](#) [Kth Largest in Stream](#) [Kth Smallest in Sorted Matrix](#) [K Min Sum Combinations](#) [Kth Smallest Fraction](#) [Sort K Sorted Array](#) [Merge K Sorted Arrays](#) [Smallest Range in K Lists](#)

## Basic Geometry Problems

[Basic Geometry](#) [Geometric Algorithms](#) [Rectangle Overlap](#) [Largest Triangle Perimeter](#) [Matrix Cells in Order](#) [Largest Triangle Area](#) [Construct Rectangle](#) [Two Collinear Points](#) [Non Collinear Points](#) [Valid Square Points](#) [Max Distance Points](#) [Total Rectangle Area](#) [Max Collinear Points](#) [Best Meeting Point](#) [Line Reflection](#) [Intersecting Lines](#) [Check Pt Inside Triangle](#)

## Two Pointer Technique

[Two Pointer Theory](#) [Sliding Window Theory](#)

### Target Sum Pair

[Two Sum - Unsorted](#) [Two Sum - Unsorted](#) [Two Sum - Sorted](#) [K Sum Pairs - Remove Pairs](#) [K Sum Pairs - Two Arrays](#) [K Sum Pairs - Unique Pairs](#) [K Sum Pairs - All Pairs](#) [K Sum Pairs - Sorted Matrix](#) [Two Sum - Closest](#) [Two Sum - Design](#) [Two Sum - Smaller](#) [Two Sum - Greater](#) [Boats to Save People](#) [Target Difference Pair](#) [K Difference Pairs](#) [Longest Difference Pair](#)

### Target Sum Triplet

[3 Sum](#) [3 Sum - Closest](#) [3 Sum - Smaller](#) [Target Sum Triplets](#) [Count Valid Triplets](#) [Valid Triangles](#) [Good Triplets](#) [Closest in 3 Arrays](#)

### Quadruplet Sum

[4 Sum - I](#) [4 Sum - Using 2 Pointer](#) [4 Sum - Using Hashmap](#) [4 Sum - II](#) [Tuples - Equal Sum](#) [Tuples - Equal Product](#) [Special Quadruplets](#) [Target Sum K Set](#)

## Two Pointer Problems

[Trapping Rain Water](#) [Container with Most Water](#) [Max Consecutive Ones](#) [Remove All Occurences](#) [Remove Duplicates Sorted - I](#) [Remove Duplicates Sorted - II](#) [Long Pressed Name](#) [Longest Consecutive Chars](#) [Count Equal Consecutive 0s 1s](#) [Happy Number](#) [Fizz Buzz](#) [Smallest String Deleting Ends](#) [Shorted Unsorted Subarray](#)

## Static Sliding Window

[Max Sum Subarray Size K](#) [Max Subarray Average Size K](#) [K Radius Subarray Averages](#) [Min Sum Subarray Size K](#) [Sliding Window Maximum](#) [Anagram Permutation](#) [Distinct Nos Sliding Window](#) [Min Swaps Group Together - I](#) [Min Swaps Group Together - II](#) [Min Swaps Group Together - III](#) [First Negative Sliding Window](#) [Substring Concatenations](#) [Max Sum Atmost 2 Unique](#) [Sliding Window Median](#)

## Dynamic Array Window

[Smallest Subarray Sum  \$\geq X\$  - I](#) [Smallest Subarray Sum  \$\geq X\$  - II](#) [Maximum Subarray Sum  \$< X\$](#)  [Count Subarrays Score  \$< K\$](#)  [Count Subarrays Product  \$< X\$](#)  [Subarrays with Max Value  \$> K\$](#)  [Subarrays Bounded Maximum](#) [Max Sum Subarray All Unique](#) [Max Sum Rectangle  \$< K\$](#)  [Sum Max Consecutive 1s](#) [K Flips](#)

## Dynamic String Window

[Minimum Window Substring - I](#) [Minimum Window Substring - II](#) [Minimum Window Subsequence](#) [Longest Substring All Unique](#) [Count Substrings All Unique](#) [Longest Substring Atmost K Unique](#) [Count Substrings Atmost K Unique](#) [Longest Substring Exact K Unique](#) [Count Substrings Exact K Unique](#) [Count Subarrays K Odd](#) [Longest Substring Atleast K Repeating](#) [Count Substrings All Chars](#) [Longest Repeating Replacement](#)

## Greedy Algorithms

[Greedy Algorithms](#)

### Meeting Rooms

[Job Sequencing](#) [Merge Intervals](#) [Insert Interval](#) [Interval Intersections](#) [Max Overlap Intervals](#) [Video Stitching](#) [Max Meetings - I](#) [Max Meetings - II](#) [Max Meetings - III](#) [Activity Selection](#) [Max Train Stoppage](#) [Disjoint Intervals](#) [Min Balloon Bursts](#) [Max Chain Length](#) [Min Meeting Rooms](#) [Min Train Platforms](#) [Car Pooling](#) [Teemo Attacking](#) [Employee Free Time](#) [My Calendar - I](#) [My Calendar - II](#) [My Calendar - III](#)

## Huffman Coding

[Weight Balanced Tree](#) [Huffman Encoding](#) [Huffman Decoding](#) [Huffman Coding](#) [Min Cost of Ropes](#) [Encode & Decode String](#)

## Max or Min in Array

[Max Product of 3 Nos](#) [Min Subset Greater Sum](#) [Smallest Range Score](#) [Least Unique Elements](#) [Min Difference 3 Moves](#) [Min Fibonacci Sum K](#) [Min Moves Equal - I](#) [Min Moves Equal - II](#) [Partition Disjoint Intervals](#) [Not Target Sum Subset](#) [Max Sum 2nd Smallest](#) [Indian Coin Change](#) [Chocolate Distribution](#) [Min & Max Candies](#) [Highest Pyramid](#) [Max Sum Rotation](#) [Minimum Jumps in Seats](#) [Max Profit Ticket Selling](#)

## Array Permutation

[Max Sum Value X Index](#) [Coordinate Compression](#) [Assign Cookies](#) [Lemonade Change](#) [Next Permutation - I](#) [Next Permutation - II](#) [Smallest Permutation](#) [Largest Permutation](#) [Largest No From Array](#) [Min Sum of 2 Numbers](#) [Arithmetic Sequence](#) [DI String Match](#) [Max Absolute Difference - I](#) [Max Absolute Difference - II](#) [Reorganize String - I](#) [Reorganize String - II](#) [Distinct Barcodes](#) [Task Scheduler](#) [Max Profit Assigning Work](#) [Shortest Job First Scheduling](#)

## Array Partitions

[Partition Labels](#) [Max Chunks to Sort - I](#) [Max Chunks to Sort - II](#) [Min Moves Unique Array](#) [Array Partition Pairs](#) [Largest Derangement](#) [Circular Tour - I](#) [Circular Tour - II](#) [Candy Distribution](#) [Kill Most Monsters](#) [Advantage Shuffle](#) [Minimize the Heights - I](#) [Minimize the Heights - II](#)

## Custom Sorting

[Median from Data Stream](#) [Minimize Cash Flow](#) [Two City Scheduling](#) [Maximum Profit IPO](#) [Height Queue Reconstruct](#) [Defense Kingdom](#) [Custom Sort String](#) [Marks of PCM](#) [Car Fleet](#) [Wine Buying](#)

[& Selling Arrange Amplifiers Biased Standings](#)

## Trees

[Tree Theory](#)

## Depth First Search

## Basics of DFS

[Constructor - Binary Tree](#) [Tree Properties](#) [Size - Binary Tree](#) [Max & Min - Binary Tree](#) [Preorder Traversal - Binary Tree](#) [Inorder Traversal - Binary Tree](#) [Postorder Traversal - Binary Tree](#) [Recursive DFS Traversals](#) [Iterative DFS Traversals](#) [Iterative DFS Traversals](#) [Searching in Binary Tree](#) [BFS vs DFS Traversals - I](#) [BFS vs DFS Traversals - II](#) [Preorder Predecessor](#) [Preorder Successor](#)

## Depth and Width

[Max Depth of Binary Tree](#) [Min Depth of Binary Tree](#) [Is Binary Tree Balanced](#) [Diameter of Binary Tree](#) [Max Width of Binary Tree - I](#) [Max Width of Binary Tree - II](#) [Vertical Width or Shadow](#)

## Node Paths Problems

[Ancestors in Binary Tree](#) [Max Node Ancestor Difference](#) [Count Turns in N2N Path](#) [All Root to Leaf Tree Paths](#) [Sum of Root to Leaf Paths](#) [Linearize Binary Tree](#) [Linked List in Binary Tree](#) [Count Good Nodes](#)

## Subtree Sum Problems

[Maximum Sum Subtree](#) [Tilt of Binary Tree](#) [Target Sum Subtrees](#) [Transform to Sum Tree](#) [Check Sum Tree](#)

## Path Sum Problems

[Has Root to Leaf Path Sum](#) [All Root to Leaf Path Sum](#) [Longest Root to Leaf Path Sum](#) [Node to Descendant Path Sum](#) [Max Leaf to Leaf Path Sum](#) [Max Node to Node Path Sum](#) [Longest Univalued Path](#)

## K Distance Nodes

[Nodes K Level Down](#) [Nodes At K Distance](#) [Burning Tree](#)

## Modify Binary Tree

[Delete Single Child Parents](#) [Merge Two Binary Trees](#) [Delete Target Leaf Nodes](#) [Delete Insufficient Nodes](#) [Delete Nodes & Return Forest](#) [Add One Row to Tree](#) [Follow Children Sum Property](#)

## Compare Trees

[Identical or Same Trees](#) [Foldable Binary Tree](#) [Isomorphic Binary Tree](#) [Symmetric or Mirror Tree](#) [Invert Binary Tree](#) [Find Cloned Tree Node](#) [Image Multiplication](#) [Duplicate Subtree - I](#) [Duplicate Subtree - II](#)

## Complete Binary Tree

[Count Complete Tree Nodes](#) [Is Binary Tree Complete](#) [Design Complete Tree](#)

## More DFS Problems

[Splitted Tree Max Edge Score](#) [Splitted Tree Max Node Score](#) [Holiday Accommodation](#) [Print Single Child Nodes](#) [Tree Coloring Game](#) [2nd Min in Special Tree](#) [Path in ZigZag Labelled Tree](#)

## Breadth First Search

[Level Order of Binary Tree](#) [Level Order Linewise](#) [Reverse Levelorder](#) [Zigzag Levelorder](#) [Alternate Levelorder](#) [Cousins in Binary Tree](#) [Maximum Level Sum](#) [Average Levelorder](#) [Are Leaves at Same Level](#) [Populate Right Pointer - I](#) [Populate Right Pointer - II](#) [Even Odd Tree](#)

## Other Tree Traversals

### Diagnol Order

[Diagonal Order - I](#) [Diagonal Order - II](#) [Diagonal Sum](#)

### Vertical Order

[Vertical Order - I](#) [Vertical Order - II](#) [Vertical Order Sum](#)

### View of Tree

[Left View of Binary Tree](#) [Right View of Binary Tree](#) [Top View of Binary Tree](#) [Bottom View of Binary Tree](#) [Boundary Traversal of Tree](#)

### Morris Traversal

[Threaded Binary Tree](#) [Convert Tree to Threaded - I](#) [Convert Tree to Threaded - II](#) [Morris Inorder Traversal](#) [Morris Preorder Traversal](#) [Morris Postorder Traversal](#)

### Construct Binary Tree

[Serialize & Deserialize Binary Tree](#) [Verify Preorder Serialization](#) [Tree from Preorder & Inorder](#) [Tree from Postorder & Inorder](#) [Tree from Preorder & Postorder](#) [Tree from Levelorder & Inorder](#) [Clone Binary Tree](#) [Binary Tree from Parent Array](#) [Bracket String from Binary Tree](#) [Binary Tree from Bracket String](#) [Ternary Expression to Binary Tree](#) [Heap Ordered Binary Tree](#)

### Lowest Common Ancestor

### LCA Variations

[LCA - Binary Tree](#) [LCA - Deepest Leaves](#) [LCA - Parent Pointers](#) [LCA - 3 Nodes](#) [LCA - K Nodes](#)  
[Shortest Distance of 2 Nodes](#)

## Sparse Table

[Sparse Table](#) [Sparse Table](#) [LCA - Using Sparse Table](#) [LCA - Using Sparse Table](#)

## Binary Lifting

[Binary Lifting - Kth Ancestor](#) [Binary Lifting - Kth Ancestor](#) [LCA - Using Binary Lifting](#) [LCA - Using Binary Lifting - I](#) [LCA - Using Binary Lifting - II](#) [Distance Queries](#) [Distance Queries](#) [LCA - Sparse Matrix](#)

## Rerooting Technique

[DP on Trees](#) [Subordinates](#) [Tree Matching](#) [Tree Distances - I](#) [Tree Distances - II](#) [House Robber - Tree](#) [Largest Independent Set](#) [Vertex Cover Tree](#) [Longest ZigZag Path](#) [Binary Tree Cameras](#) [Distribute Coins](#) [Min Time to Collect Apples](#) [Min Cost Tree From Leaf](#)

## N Ary or Generic Tree

[N Ary Tree Theory](#) [N Ary Tree Problems](#) [Preorder DFS - Nary](#) [Postorder DFS - Nary](#) [Height of Nary Tree](#) [Diameter of Nary Tree](#) [Mirror of Nary Tree](#) [Levelorder BFS - Nary](#) [Nary to Binary Tree](#) [Serialize Deserialize Nary](#) [Lockable Tree](#) [Lockable Tree](#)

## Binary Search Tree

## BST Basics

[Binary Search Tree Theory](#) [Insertion in BST](#) [Deletion in BST](#) [Searching in BST](#) [Minimum Node in BST](#) [Maximum Node in BST](#) [Handle Duplicates in BST](#) [DFS Generic Solver in BST](#) [LCA in BST](#)

## Inorder Traversal

[Inorder Pred/Successor in BST](#) [Inorder Successor in BST](#) [Kth Smallest Element in BST](#) [Kth Largest Element in BST](#) [Greater Sum Tree from BST](#) [Min Difference in BST](#) [Mode in BST](#) [Recover BST](#) [Union of BSTs](#)

## Range Problems

[Print Nodes in Range](#) [Count Nodes in Range](#) [Range Sum of BST](#) [Trim BST](#) [Ceil Element in BST](#) [Floor Element in BST](#) [Closest Element in BST](#) [K Closest Elements in BST](#) [Split BST](#)

## Construct BST

[Construct BST From Inorder](#) [Construct BST From Preorder](#) [Construct BST From Postorder](#) [Construct BST From Levelorder](#) [Convert Sorted SLL to BST](#) [Convert BST to Sorted SLL](#) [Convert BST To Sorted DLL](#) [Convert BST To Sorted CDLL](#) [Convert Sorted DLL to BST](#) [Convert Sorted DLL to BST](#) [Serialize & Deserialize BST](#) [Balance a BST](#) [Depth of BST from Array](#)

## Check for BST

[Validate BST](#) [Validate BST](#) [Largest BST Subtree](#) [Max Sum BST Subtree](#) [Check Dead End in BST](#) [Verify Preorder of BST](#) [Identical BST](#) [Identical BST -  \$O\(n^2\)\$](#)  [Identical BST -  \$O\(n\)\$](#)

## BST Iterator

[BST Iterator - I](#) [BST Iterator - II](#) [Iterator & Iterable](#) [Forward Iterator in BST](#) [Backward Iterator in BST](#) [Target Sum Pair in BST - I](#) [Target Sum Pair in BST - II](#) [Median of BST](#) [Merge Two BSTs](#)

## Dynamic Programming

[DP Theory](#)

## Fibonacci Sequence



[Fibonacci Number](#) [Tribonacci Number](#) [Valid Binary Strings](#) [Arrange Buildings](#) [Decode Ways](#) [A+B+C+ Subsets](#) [Tiling - 2 X 1 Tiles](#) [Tiling - M X 1 Tiles](#) [Dominos & Trominos](#) [Count Friend Pairing](#) [Distinct Subsequences](#) [Ugly Numbers](#) [Super Ugly Numbers](#) [Count Derangements](#) [Assembly Line Scheduling](#) [Assembly Line Scheduling](#) [Weighted Job Scheduling - I](#) [Weighted Job Scheduling - II](#) [Max Taxi Earnings](#) [Min Flips Sort Binary](#) [Min Swaps Increasing Sets](#)

## Climbing Stairs

[Climb Stairs - I](#) [Climb Stairs - II](#) [Climb Stairs - III](#) [Climb Stairs - III](#) [Climb Stairs - IV](#) [Jump Game - I](#) [Jump Game - II](#) [Jump Game - III](#) [Jump Game - All Paths](#) [Jump Game - All Paths](#) [Frog Jump - I](#) [Frog Jump - II](#) [Min Steps to 1](#) [Min Taps to Water](#) [2 Keys Keyboard](#) [4 Keys Keyboard](#)

## House Robber

[House Robber - I](#) [House Robber - II](#) [House Robber - III](#) [Paint House - I](#) [Paint House - II](#) [Paint Fence - I](#) [Paint Fence - II](#)

## Knapsack Problem

[Fractional Knapsack](#) [0-1 Knapsack](#) [0-1 Knapsack - Print Ways](#) [0-1 Knapsack - Print Ways](#) [Unbounded Knapsack](#) [Rod Cutting Max Sum](#) [Min Cost for Tickets](#) [Min Cost to Fill Bag](#) [Flip Array Signs](#) [1s & 0s Binary String](#)

## Coin Change Problem

[Minimum Coin Change](#) [Coin Change - Combinations](#) [Coin Change - Permutations](#) [Coin Change Analysis](#)

## Buy & Sell Stock

[1. One Transaction](#) [2. Infinite Transactions](#) [3. Transaction Fees](#) [4. Cooldown](#) [5. Two Transactions](#) [6. K Transactions](#)

## Target Sum Subset

[Check Target Sum Subset](#) [Count Target Sum Subset](#) [Print All Target Sum Subsets - I](#) [Print All Target Sum Subsets - II](#) [Equal Sum Partition](#) [Target Diff Partition](#) [Equal Average Partition](#) [K Equal Sum Partition](#) [K Partitions - I](#) [K Partitions - II](#) [Bell Numbers](#) [Tug of War - Equal Size](#) [Tug of War - Diff Size](#)

## DP on Grid

[Min Path Sum in Maze](#) [Min Path Sum - All Paths](#) [Min Path Sum in Triangle](#) [Minimum Falling Path Sum](#) [Goldmine All Paths](#) [Unique Paths - I](#) [Unique Paths - II](#) [Knight's Probability Chess](#) [Keypad Problem - DP](#) [Dungeon Game](#) [Cherry Pickup - I](#) [Cherry Pickup - II](#) [Maximal Square](#) [Square Submatrix All 1s](#) [Out of Boundary Paths](#) [Largest Bordered Square](#) [Knight Dialer](#) [Largest Plus Sign](#) [Divide Chocolate Bar](#)

## Pascal Triangle

[Pascal Triangle - I](#) [Pascal Triangle - II](#) [Binomial Coeff \(nCr\)](#) [Permutation Coeff \(nPr\)](#) [Water Overflow](#)

## LIS Problems

[Longest Increasing Subset](#) [Longest Increasing Subarray](#) [Print All LIS](#) [Count All LIS](#) [Longest Increasing Subarray](#) [Max Sum Increasing Subarray](#) [Longest Bitonic Subset](#) [Max Sum Bitonic Subset](#) [Longest Bitonic Subarray](#) [Building Bridges](#) [Russian Doll Envelopes](#) [Perfect Squares](#) [Count AP Subarray](#) [Count AP Subsets](#) [Longest AP Subarray](#) [Longest AP Subset](#) [Longest Wiggle Subset](#) [Highway Billboards](#) [Box Stacking - I](#) [Box Stacking - II](#) [Largest Divisible Subset](#) [LCIS \(LCS + LIS\)](#) [Longest String Chain](#)

## LCS Problems

### Longest Common Subset

[Longest Common Subset](#) [Uncrossed Lines](#) [Print Any LCS](#) [Print All LCS](#) [Longest Common Substring](#) [Increase LCS by 1](#) [LCS of 3 Strings](#)

### Longest Duplicate Subset

[Longest Duplicate Subset](#) [Longest Duplicate Substring - I](#) [Longest Duplicate Substring - II](#)

## Palindromic Subsets

[Longest Palindromic Subset](#) [Count Palindromic Subset - I](#) [Count Palindromic Subset - II](#) [Min Deletions for Palindrome](#) [Min Insertions for Palindrome](#) [K Palindromic String](#)

## Palindromic Substrings

[Count Palindromic Substrings Using Dynamic Programming](#) [Using Expand Around Center](#) [Print All Palindromic Substrings](#) [Distinct Palindromic Substrings Using Dynamic Programming](#) [Using Manacher's Algorithm](#) [Longest Palindromic Substring Using Dynamic Programming](#) [Using Expand Around Center](#) [Manacher's Algorithm](#)

## Expression Matching

[Wildcard Matching](#) [Regular Expression Matching](#) [Edit Distance](#) [Min Cost for Identical Strings](#) [Min Delete Operations](#) [Min ASCII Delete Sum](#) [Interleaving String](#) [Distinct Transformations](#) [Min Insertions and Deletions](#)

## Shortest Common Superset

[Shortest Common Superset - I](#) [Shortest Common Superset - II](#) [Shortest Uncommon Subset](#)

## Catalan Numbers

[Nth Catalan Number](#) [Nth Catalan - 3 Solns](#) [Catalan No Applications](#) [Catalan No Applications](#) [Unique BSTs - I](#) [Unique BSTs - II](#) [Count Balanced Paranthesis](#) [Count of Mountain Ranges](#) [Dyck Paths](#) [Dyck Words](#) [Non Intersecting Chords](#) [Count Handshakes](#) [Count Triangulations](#) [Min Score Triangulation](#)

## Kadane's Algorithm

[Maximum Subarray Sum](#) [Max Absolute Sum Subarray](#) [Max Sum Circular Subarray](#) [K Concatenation](#) [Max Sum](#) [Maximum Product Subarray](#) [Maximum Sum Submatrix](#) [Max Sum](#)

[Subarray >= K Size](#) [Best Sightseeing Pair](#) [Max Difference of 0s 1s](#) [Max Sum 2 Non-Overlap Subarrays](#) [Max Sum 3 Non-Overlap Subarrays](#) [Max Sum 3 Non-Overlap Subarrays](#) [Max Sum K Non-Overlap Subarrays](#)

## Optimal Game Strategy

[Predict the Winner](#) [Stone Game](#) [Optimal Game Strategy](#) [Wine Selling Problem](#) [Egg Drop - 2 Eggs](#) [Egg Drop - K Eggs](#) [Super Egg Drop](#) [Stone Game - II](#) [Stone Game - III](#) [Stone Game - IV](#) [Stone Game - V](#) [Stone Game - VII](#) [Stone Game - VIII](#)

## MCM Problems

## Palindromic Partitioning

[Palindrome Partitioning - I](#) [Palindrome Partitioning - II](#) [Palindrome Partitioning - III](#) [Palindrome Partitioning - IV](#)

## Matrix Chain Multiplication

[Matrix Chain Multiplication](#) [Printing Brackets](#) [Parsing Boolean Expression](#) [Boolean Parenthesization](#) [Optimal Binary Search Tree](#) [Partition array for Max Sum](#) [Burst Balloons](#) [Scramble String](#) [Rectangle Cutting](#) [Min Cost to Merge Stones](#) [Min Cost to Cut Stick](#) [Min & Max Values with \\* +](#)

## Word Break

[Check Word Break](#) [Concatenated Words](#) [Word Break - Backtracking](#) [Word Break - DP](#) [Minimum Word Break](#) [Text Justification - Greedy](#) [Text Justification - DP](#)

## Bit Manipulation

## Number System

[Number System Basics](#) [Decimal to Any Base](#) [Decimal to Any Base](#) [Decimal to Base 7](#) [Decimal to Hexadecimal](#) [Decimal to -2 Base](#) [Any Base to Decimal](#) [Convert Binary to Decimal](#) [Any Base to Any Base](#) [Any Base to Any Base](#) [Any Base Addition](#) [Any Base Addition](#) [Add Binary Numbers](#) [Add -2 Base Numbers](#) [Any Base Subtraction](#) [Any Base Subtraction](#) [Any Base Multiplication](#) [Any Base Multiplication](#)

## Bitmasking Basics

[Bit Manipulation Theory](#) [Bit Manipulation Basics](#) [Bitmasking Tricks](#) [Bitmasking Tricks](#) [Binary Representation](#) [1s Complement](#) [2s Complement](#) [Check Bits](#) [Check Odd or Even](#) [Set Bits](#) [Unset or Clear Bits](#) [Toggle Bits](#) [Swap 2 Nos Using Bits](#) [Power Set Using Bits](#) [Power Set Using Bits](#)

## Hamming Weight

[Find Rightmost Set Bit](#) [Rightmost Setbit Mask](#) [Set Rightmost Unset Bit](#) [Hamming Weight - I](#) [Kernighan's Algorithm](#) [Hamming Weight - II](#) [Hamming Weight - II](#) [Hamming Distance](#) [Total Hamming Distance](#) [Total Hamming Distance](#) [Prime Set Bits](#) [Steps Reduce to 0](#)

## Bitset Problems

[Min Bit Flips to Convert No](#) [Min Bit Flips to Convert No](#) [Same No of Set Bits](#) [Same No of Set Bits](#) [Copy Set Bits in Range](#) [Copy Set Bits in Range](#) [Reverse Bits](#) [Reverse Bits](#) [Swap Odd Even Bits](#) [Swap Odd Even Bits](#) [Swap Nibbles in Byte](#) [Check Power of 2](#) [Check Power of 2](#) [Highest Power of 2](#) [Check Power of 3](#) [Check Power of 4](#)

## Gray Codes

[Generate Gray Codes - I](#) [Generate Gray Codes - II](#) [Generate Gray Codes](#) [Gray - Binary Conversion](#) [Binary to Gray Conversion](#) [Gray to Binary Conversion](#)

## Single Number

[Single Number - I](#) [Single Number - I](#) [Single Number - II](#) [Single Number - II](#) [Single Number - III](#) [Single Number - III](#)

## XOR Problems

[Triplets with = XOR](#) [Triplets with = XOR](#) [XOR of Sum of Pairs](#) [XOR of Sum of Pairs](#) [Min Xor Pair](#) [Min Xor Pair](#) [Xor Subarray Queries](#) [Max XOR in Range](#) [XOR of All Subarrays](#) [Count Pairs = XOR](#) [Bitwise AND of Range](#)

## More Bits Problems

[Josephus Special Case](#) [Josephus Special Case](#) [Divide Integers w/o Operators](#) [Divide Integers w/o Operators](#) [Add Integers w/o Operators](#) [Valid Words for Puzzle](#) [Valid Words for Puzzle](#) [Integer Replacement](#) [Integer Replacement](#) [Is No Multiple of 3](#) [Is No Multiple of 3](#) [UTF - 8](#) [UTF - 8](#) [Encoding](#) [Palindromic Binary No](#) [Palindromic Binary No](#) [Binary Substrings 1 to N](#) [Count Steps to Reduce to 1](#)

## Graphs

[Graph Theory](#)

### Basic BFS/DFS

[DFS](#) [DFS - Multisolver](#) [DFS - Iterative](#) [DFS - Applications](#) [BFS](#) [BFS - Applications](#) [Has Path \(IB\)](#) [Print All Paths](#) [Time to Inform Employees](#) [Min Swaps to Sort Array](#) [Clone Graph](#) [Hamiltonian Path](#) [Hamiltonian Cycle](#) [Hamiltonian Cycle](#) [Pacific Atlantic Water Flow](#) [Surrounded Regions](#) [Reorder Routes](#) [Water Jug Problem](#) [Smallest Multiple with 0 & 1](#)

## Euler Path & Circuit

[Euler Path & Circuit](#) [Seven Bridges of Königsberg](#) [Euler Path - Undirected](#) [Euler Path - Directed](#) [Reconstruct Itinerary](#)

## Connected Components

[Number of Provinces](#) [Make Graph Connected](#) [Unreachable Pair Nodes](#) [Minimize Malware Spread](#) [Number of Islands](#) [Number of Enclaves](#) [Number of Distinct Islands](#) [Island Perimeter](#) [Island Area](#) [Flood Fill](#) [SCC - Kosaraju Algo](#) [Kosaraju - DFS](#) [Kosaraju - BFS](#) [SCC - Tarjan Algo](#) [Tarjan Algorithm](#)

## Cycle Detection

[Cycle in Undirected Graph](#) [Undirected - DFS](#) [Undirected - BFS](#) [Cycle in Directed Graph](#) [Directed - DFS - 2 Arrays](#) [Directed - DFS - Coloring](#) [Directed - BFS](#) [Graph - Valid Tree](#) [Eventual Safe States](#) [Longest Cycle Length](#) [Print All Cycles](#)

## Topological Sort

[Topological Sort](#) [Topological Sort - DFS](#) [Topological Sort - BFS](#) [Course Schedule - I](#) [Course Schedule - II](#) [Parallel Courses](#) [Alien Dictionary - I](#) [Alien Dictionary - II](#) [Mother Vertex](#) [Assign Directions - DAG](#) [Minimum Height Trees](#)

## Graph Coloring

[Bipartite Graph](#) [Bipartite Graph - BFS](#) [Bipartite Graph - DFS](#) [Possible Bipartition](#) [Two Clique Problem](#) [M Coloring Problem](#) [Graph Coloring](#)

## Shortest Path

## Unweighted Graph

[Unweighted Graph](#) [Unweighted Graph](#) [Spread of Infection](#) [Rotten Oranges](#) [Snake and Ladder](#) [Word Ladder - I](#) [Word Ladder - II](#) [Nearest 0 in Binary Matrix](#) [Shortest Path in Binary Matrix](#) [Min Steps by Knight](#) [Shortest Bridge](#) [Shortest Path Visiting All Nodes](#) [Farthest Land Possible](#) [Sliding Puzzle](#) [K Similar Strings](#) [0-1 BFS](#) [Minimum Edges Reversals](#) [Min Cost for Valid Path](#) [Trapping Rain Water 3D](#) [Jump Game - III](#) [Jump Game - IV](#)

## Dijkstra's Algorithm

[Network Delay Time](#) [Dijkstra Algorithm](#) [Min Cost Path in Grid](#) [Path with Max Probability](#) [Path with Min Effort](#) [Count Shortest Paths](#) [Bus Routes](#)

## Bellman Ford Algo

[Bellman Ford Algorithm](#) [Bellman Ford Algorithm](#) [Negative Weight Cycle](#) [Cheapest Flights K Stops](#)

## **Floyd Warshall Algo**

[Floyd Warshall Algorithm](#) [Floyd Warshall Algorithm](#) [City with Min Neighbours](#) [Shortest Paths Queries](#)

## **Directed Acyclic Graph**

[Shortest Path in DAG](#) [Longest Path in DAG](#) [Longest Path - Tree](#) [Longest Increasing Path Matrix](#) [Jump Game - V](#) [Shortest Path in Multistage Graph](#)

## **Disjoint Set Union (DSU)**

[Disjoint Set Union](#) [Disjoint Set Union \(DSU\)](#) [No of Island Queries](#) [Smallest Equivalent String](#) [Has Path Queries](#) [Similar String Groups](#) [Smallest String with Swaps](#) [Satisfiability of Eqn](#) [Redundant Connection - I](#) [Redundant Connection - II](#) [GCD Threshold Connectivity](#) [Evaluate Division](#) [Colorful Array](#) [Remove Max Edges](#) [Accounts Merge](#) [Rank Transform of Matrix](#) [Most Stones Removed](#) [Making Large Island](#) [Swim in Rising Water](#)

## **Minimum Spanning Tree**

[Minimum Spanning Tree](#) [Prim's Algorithm](#) [Kruskal's Algorithm](#) [Min Cost to Connect Points](#) [Optimize Water Distribution](#)

## **Articulation Pt & Bridges**

[Euler Tour of Tree](#) [Edge Classification](#) [Articulation Points](#) [Articulation Points](#) [Critical Connections](#) [Bridges - Offline Algo](#) [Bridges - Online Algo](#)

## **Trie or Prefix Tree**



[Trie Theory](#) [Trie Applications](#) [Implement Trie/Prefix Tree - I](#) [Implement Trie/Prefix Tree - II](#)  
**Prefix Tree Problems**

[Design Add Search Word](#) [Weighted Prefix Search](#) [Longest Common Prefix](#) [Longest Word in Dictionary](#) [Implement Magic Dictionary](#) [String Searching Queries](#) [Design Search Suggestions](#) [Word Search - II](#) [Word Boggle](#) [Replace Words with Prefix](#) [Prefix & Suffix Search](#) [Palindrome Pairs](#) [Shortest Unique Prefix](#) [Count Distinct Substrings](#)

## Binary Trie Problems

[Maximum XOR Pair - I](#) [Maximum XOR Pair - II](#) [XOR Pairs in Range](#) [Unique Rows in 01 Matrix Subarrays with XOR < K](#)

## Competitive Programming

[CP Handbook](#) [CSES Problem Set](#) [CP Algorithms](#)  
**Template**

[Setting IDE Java](#) [Setting IDE C++](#) [Fast I/O Java](#) [Fast I/O C++](#) [Java Tricks](#) [C++ Tricks](#)

## String Pattern Matching

[Pattern Matching](#) [Search Occurrence](#) [First Occurrence](#) [Java Regular Expressions](#) [Longest Prefix Suffix](#) [Longest Palindrome Prefix](#) [KMP Algorithm](#) [KMP Algorithm](#) [Repeated String Pattern - I](#) [Repeated String Pattern - II](#) [KMP & Z Algorithm](#) [Shortest Palindrome](#) [String Hashing](#) [Rolling Hash Function](#) [Rabin Karp Algorithm](#) [Rabin Karp Algorithm](#) [Rabin Karp Algorithm](#) [Distinct Echo Substrings](#) [Repeated DNA Sequences](#) [Z Algorithm](#) [Z Algorithm](#) [Compressed Trie](#) [Suffix Trees](#) [Suffix Trees](#)

## Advanced Mathematics

[O.E.I.S Series](#) [Basic Combinatorics](#)

## Advanced Number Theory

[Euler's Totient Function](#) [Euler's Totient Function](#) [Euler's Totient Function](#) [Chinese Remainder Theorem](#) [Chinese Remainder Theorem](#) [Pigeonhole Principle](#) [Pigeonhole Principle](#)

## Large Exponentiation

[Big Power - Large A](#) [Big Power - Large A](#) [Big Power - Large A](#) [Big Power - Large B](#) [Big Power - Large B](#) [Super Power](#) [Super Power](#) [Power of Power](#) [Power - Big Integers](#) [Modular Factorial](#) [Wilson's Theorem](#) [Wilson's Theorem](#) [Factorials Again](#) [Product of Factorials](#)

## Fibonacci Numbers

[Fibonacci Numbers](#) [Linear Recurrence Relation](#) [Facts About Fibonacci](#) [Matrix Exponentiation](#) [Matrix Exponentiation](#) [Generalized Fibonacci](#) [Modified Fibonacci](#) [Cassini Identity](#) [Check Fibonacci](#) [GCD & Fibonacci](#) [Nth Fibonacci Last Digit](#) [Nth Digit of Fibonacci](#) [Nth Non Fibonacci](#) [Nth Even Fibonacci](#) [Fibonacci Sum](#) [Throwing Dice](#) [Dice Roll Expectation](#) [Recursive Sequence - I](#) [Recursive Sequence - II](#)

## Probability Problems

[Birthday Paradox](#) [Probability Theory](#) [Mathematical Expectation](#) [Mathematical Expectation](#) [Coupon Collector Problem](#) [Dice Probability](#) [Candy Lottery](#) [Inversion Probability](#) [Moving Robots](#)

## Game Theory

[Game Theory Notes](#) [Game Theory Notes](#) [Game Theory Notes](#) [Sprague Grundy Theorem](#) [Finders Keeper](#) [Game of Nim](#) [QCJ3 Game](#) [Stone Game - VI](#) [Maximum Coins](#) [Sum Game](#) [Remove Same Colors](#) [CSES - Stick Game](#) [CSES - Nim Game I](#) [CSES - Nim Game II](#) [CSES - Stair Game](#) [CSES - Grundy's Game](#) [CSES - Another Game](#)

## Advanced Searching

[Jump Search](#) [Step Array Search](#) [Exponential Search](#) [Exponential Search](#) [Unbounded Binary Search](#) [Searching in Infinite Sorted](#) [Interpolation Search](#) [Interpolation Search](#) [Ternary Search](#)

[Ternary Search](#) [Ternary Search](#)

## **Advanced Linked Lists**

### **XOR List**

[XOR List](#) [XOR LL](#) [XOR LL - Code](#) [XOR List - I](#) [XOR List - II](#)

### **Skip List**

[Design Skip List](#) [Introduction](#) [Insertion](#) [Search & Deletion](#) [Skip List](#)

### **Self Organizing List**

[Introduction](#) [1. Move to Front Method](#) [2. Count Method](#) [3. Tranpose Method](#) [Self Organizing List](#)

### **Unrolled Linked List**

[Introduction](#) [Introduction](#) [Insertion](#) [Insertion](#)

## **Advanced Trees**

### **AVL Tree**

[AVL Tree Theory](#) [AVL Tree Theory](#) [AVL Tree Insertion](#) [AVL Tree Insertion](#) [AVL Tree Deletion](#)  
[AVL Tree Deletion](#)

### **B & B+ Tree**

[B & B+ Tree](#) [B Tree Theory](#) [B Tree Insertion](#) [B Tree Deletion](#) [B+ Tree Theory](#) [B+ Tree Insertion](#)  
[B+ Tree Deletion](#)

## **M Way Tree**

[2-3 Trees](#) [2-3 Trees](#) [2-3-4 Trees](#) [2-3-4 Trees](#) [M Way Tree](#) [M Way Tree - I](#) [M Way Tree - II](#)

## **Red Black Trees**

[Red Black Tree](#) [RB Tree Theory](#) [RB Tree Insertion](#) [RB Tree Deletion](#) [Augmented DS](#)  
[Augmented DS](#) [Interval Trees](#) [Interval Trees](#)

## **Splay Trees**

[Splay Tree Theory](#) [Splay Tree Theory](#) [Splay Tree - Searching](#) [Splay Tree - Insertion](#) [Splay Tree - Insertion](#) [Splay Tree - Deletion](#) [Splay Tree - Deletion I](#) [Splay Tree - Deletion II](#)

## **Range Query**

### **Range Query Theory**

[Segment Tree Theory](#) [Segment Tree Theory](#) [Segment Tree Theory](#) [Lazy Propagation](#) [Persistent Segment Tree](#) [Fenwick Tree Theory](#) [Fenwick Tree Theory](#) [Square Root Decomposition](#) [Square Root Decomposition](#) [Segment Tree Playlist](#) [Range Query Playlist](#)

### **Range Sum Query**

[Range Sum Query - 1D](#) [Range Sum Query - 2D](#) [RSQ - Point Update 1D](#) [RSQ - Point Update 2D](#) [RSQ - Range pdates](#) [Count Even in Range](#) [Squares Sum Range Query](#)

### **Range Minimum Query**

[Range Minimum Query](#) [Range Minimum Query](#) [RMQ - Point Updates](#) [RMQ - Range Updates](#)

## Subarray Sum Queries

[Max Subarray Sum Queries](#) [Max Subarray Sum Range - I](#) [Max Subarray Sum Range - II](#) [Max Pair Sum Range Queries](#) [Max Prefix Sum Queries - I](#) [Max Prefix Sum Queries - II](#)

## Range Query Problems

[Inversion Count](#) [Range XOR Queris](#) [Distinct Value Queries](#) [Value in Range Queries](#) [Value > K Queries - I](#) [Value > K Queries - II](#) [Chocolate & Sweetnes](#) [AND Rounds](#) [Design Tetris](#) [Frequency Range Queries](#) [Smallest Subarray GCD](#) [Palindrome Substring Queries](#) [Increasing Triplet Subsets](#) [Binary Flip Queries](#) [Card Trick](#) [Niceday of Competitors](#)

## Advanced Geometry

[Computational Geometry](#) [Line Sweep Technique](#) [Convex Polygon](#) [Convex Hull](#) [Erect the Fence](#) [Jarvis Algorithm](#) [Graham Scan Algorithm](#) [Graham Scan Algorithm](#)

## Digit Dynamic Programming

[Digit Dynamic Programming](#) [Digit Dynamic Programming](#) [Digit DP - Tight Constraint](#) [Digit DP - Leading Constraint](#) [Nos in Range with Sum Digits](#) [Count Numbers from Given Set](#) [Digit Sum in Numbers](#) [Digit Sum in Numbers](#) [Count 1s in Nos <= N](#) [Digit Count in Range](#) [Nos with No Equal Digits](#) [Nos with No Equal Digits](#) [Nos without Consecutive 1s](#) [Nos with Repeated Digits](#) [N Digit Stepping Nos](#)

## DP With Bitmasking

[Subsets Using Bitmasking](#) [DP with Bitmasking](#) [DP with Bitmasking](#) [DP with Bitmasking](#) [Travelling Salesman Problem](#) [TSP - Backtracking](#) [TSP - Backtracking](#) [TSP - Dynamic Programming - I](#) [TSP - Dynamic Programming - II](#) [TSP - Dynamic Programming](#) [Smallest Sufficient Team](#) [Smallest Sufficient Team](#) [Ways to wear Diff Hats](#) [Max Students in Exam](#) [Shortest Superstring](#) [Sum Over Subset \(SOS\)](#)

## Meet in the Middle

[Meet in the Middle](#) [Meet in the Middle](#) [Meet in the Middle](#) [Target Sum Subset](#) [Sum of Four Values](#) [Subset Sum in Range](#)

## Network Flow Algorithm

[Network Flow](#) [Max Flow Problem](#) [Max Flow Problem](#) [Ford Fulkerson Algo](#) [Dinic's Algorithm](#) [Minimum Cut](#) [Minimum Cut](#) [Max Bipartite Matching](#) [Max Bipartite Matching](#)

## System Design

[Low Level Design](#) [OOPS interview](#)

## Java Advanced

[UML - Tutorialspoint](#) [UML - Java T Point](#) [LLD Codes](#) [Real World Problems](#) [Design Patterns - Pep](#)

## Multithreading

[Print in Order](#) [Print FooBar Alternately](#) [Print Zero Even Odd](#) [Building H2O](#) [Fizz Buzz](#) [Multithreaded](#) [The Dining Philosophers](#)

## Design Principles

[Design Principles - Definitions](#) [SOLID Principles - OODesign](#) [SOLID Principles - Baeldung](#) [SOLID Principles - FreeCodeCamp](#)

## Design Patterns

[Design Patterns - Scaler Topics](#) [Design Patterns - OODesign](#) [Design Patterns - Tutorialspoint](#) [Design Patterns - JavaTPoint](#) [Design Patterns - DigitalOcean](#)

## Design Problems

## Leetcode Problems

[Design Parking Lot](#) [Design Parking Lot](#) [Design Ordered Stream](#) [Design ATM Machine](#) [Design Tiny URL](#) [Design Twitter](#) [Design Tweet Counter](#) [Design Online Election](#) [Design Timeseries Database](#) [Design Food Rating System](#) [Design BitSet](#) [Design Bank System](#) [Design Browser History](#) [Design No Container System](#) [Stock Price Fluctuation](#) [Seat Reservation Manager](#) [Authentication Manager](#) [Railway Underground System](#) [Design Book My Show](#) [Design Movie Rental App](#) [Design Text Editor](#) [Design Interval Tree](#) [Design O1 Data Structure](#)

## Design Parser

[Design Peeking Iterator](#) [Flatten Nested List Iterator](#) [Parse Nested List](#) [RLE Iterator](#) [Crawler Log Folder](#) [Simplify Path](#) [Remove Comments](#)

## Leetcode Locked

[Design Tic Tac Toe](#) [Design Hit Counter](#) [Ternary Expression Parser](#) [Design Candy Crush](#) [Design Snake Game](#) [Design Leaderboard](#) [Design File System](#) [Design In Memory File System](#) [Design Excel Sum Formula](#) [Design Log Storage System](#) [Design Logger Rate Limiter](#) [Most Recently Used Queue](#) [Design Deck of Cards](#) [Design Call Center](#)

## Machine Coding Round

[Design Parking Lot](#) [Design Splitwise App](#) [Design Chess Validator](#) [Design Snake and Ladder](#) [Design 2048 Game](#) [Design Tic Tac Toe](#) [Design Library Management](#) [Design Trello](#) [Design In Memory Cache](#) [Design Distributed Queue](#)

## High Level Design

[HLD - Donne Martin Primer](#) [HLD - GeeksForGeeks](#) [HLD - Guided Path](#) [HLD - Gaurav Sen](#) [HLD - Blogs](#) [System Design Series](#)

## Databases

[SQL vs NoSQL Database](#) [Chosing Right Database - I](#) [Chosing Right Database - II](#) [Database](#)

[Scaling Patterns](#) [Scaling Databases](#) [In Memory Database](#) [Graph Database](#) [Database Indexing](#)  
[Master Slave Database](#) [Master-Slave vs Master-Master](#) [ACID vs BASE Properties](#)

## **InterviewBit HLD**

[Design Cache](#) [Sharding Database](#) [Highly Available Database](#) [Highly Consistent Database](#)  
[Design URL Shortener](#) [Design Web Search](#) [Design Whatsapp](#) [Design Twitter](#)