

Topic: Problem:

- Array [Reverse the array](#)
- Array [Find the maximum and minimum element in an array](#)
- Array [Find the "Kth" max and min element of an array](#)
- Array [Given an array which consists of only 0, 1 and 2. Sort the array](#)
- Array [Move all the negative elements to one side of the array](#)
- Array [Find the Union and Intersection of the two sorted array](#)
- Array [Write a program to cyclically rotate an array by one.](#)
- Array [find Largest sum contiguous Subarray \[V. IMP\]](#)
- Array [Minimise the maximum difference between heights \[V.IMP\]](#)
- Array [Minimum no. of Jumps to reach end of an array](#)
- Array [find duplicate in an array of N+1 Integers](#)
- Array [Merge 2 sorted arrays without using Extra space.](#)
- Array [Kadane's Algo \[V.V.V.V.V IMP\]](#)
- Array [Merge Intervals](#)
- Array [Next Permutation](#)
- Array [Count Inversion](#)
- Array [Best time to buy and Sell stock](#)
- Array [find all pairs on integer array whose sum is equal to given value](#)
- Array [find common elements In 3 sorted arrays](#)
- Array [Rearrange the array in alternating positive and negative elements](#)
- Array [Find if there is any subarray with sum equal to 0](#)
- Array [Find factorial of a large number](#)
- Array [find maximum product subarray](#)
- Array [Find longest consecutive subsequence](#)
- Array [Given an array of size n and a number k, find all elements whose frequency is more than k](#)
- Array [Maximum profit by buying and selling a share at most twice](#)
- Array [Find whether an array is a subset of another array](#)
- Array [Find the triplet that sum to a given value](#)
- Array [Trapping Rain water problem](#)
- Array [Chocolate Distribution problem](#)
- Array [Smallest Subarray with sum greater than a given value](#)

Array [Three way partitioning of an array around a given value](#)
Array [Minimum swaps required bring elements less equal K to](#)
Array [Minimum no. of operations required to make an array p](#)
Array [Median of 2 sorted arrays of equal size](#)
Array [Median of 2 sorted arrays of different size](#)

Matrix [Spiral traversal on a Matrix](#)
Matrix [Search an element in a matrix](#)
Matrix [Find median in a row wise sorted matrix](#)
Matrix [Find row with maximum no. of 1's](#)
Matrix [Print elements in sorted order using row-column wise s](#)
Matrix [Maximum size rectangle](#)
Matrix [Find a specific pair in matrix](#)
Matrix [Rotate matrix by 90 degrees](#)
Matrix [Kth smallest element in a row-column wise sorted matrix](#)
Matrix [Common elements in all rows of a given matrix](#)

String [Reverse a String](#)
String [Check whether a String is Palindrome or not](#)
String [Find Duplicate characters in a string](#)
String [Why strings are immutable in Java?](#)
String [Write a Code to check whether one string is a rotation of](#)
String [Write a Program to check whether a string is a valid shuffle](#)
String [Count and Say problem](#)
String [Write a program to find the longest Palindrome in a string](#)
String [Find Longest Recurring Subsequence in String](#)
String [Print all Subsequences of a string.](#)
String [Print all the permutations of the given string](#)
String [Split the Binary string into two substring with equal 0's and](#)
String [Word Wrap Problem \[VERY IMP\].](#)
String [EDIT Distance \[Very Imp\]](#)

String [Find next greater number with same set of digits. \[Very](#)
String [Balanced Parenthesis problem.\[Imp\]](#)
String [Word break Problem\[Very Imp\]](#)
String [Rabin Karp Algo](#)
String [KMP Algo](#)
String [Convert a Sentence into its equivalent mobile numeric l](#)
String [Minimum number of bracket reversals needed to make](#)
String [Count All Palindromic Subsequence in a given String.](#)
String [Count of number of given string in 2D character array](#)
String [Search a Word in a 2D Grid of characters.](#)
String [Boyer Moore Algorithm for Pattern Searching.](#)
String [Converting Roman Numerals to Decimal](#)
String [Longest Common Prefix](#)
String [Number of flips to make binary string alternate](#)
String [Find the first repeated word in string.](#)
String [Minimum number of swaps for bracket balancing.](#)
String [Find the longest common subsequence between two st](#)
String [Program to generate all possible valid IP addresses from](#)
String [Write a program to find the smallest window that conta](#)
String [Rearrange characters in a string such that no two adja](#)
String [Minimum characters to be added at front to make strin](#)
String [Given a sequence of words, print all anagrams together](#)
String [Find the smallest window in a string containing all chara](#)
String [Recursively remove all adjacent duplicates](#)
String [String matching where one string contains wildcard cha](#)
String [Function to find Number of customers who could not ge](#)
String [Transform One String to Another using Minimum Numb](#)
String [Check if two given strings are isomorphic to each other](#)
String [Recursively print all sentences that can be formed from](#)

Searching & S [Find first and last positions of an element in a sorted ar](#)

Searching & S [Find a Fixed Point \(Value equal to index\) in a given array](#)

Searching & S [Search in a rotated sorted array](#)

Searching & S [square root of an integer](#)

Searching & S [Maximum and minimum of an array using minimum nu](#)

Searching & S [Optimum location of point to minimize total distance](#)

Searching & S [Find the repeating and the missing](#)

Searching & S [find majority element](#)

Searching & S [Searching in an array where adjacent differ by at most k](#)

Searching & S [find a pair with a given difference](#)

Searching & S [find four elements that sum to a given value](#)

Searching & S [maximum sum such that no 2 elements are adjacent](#)

Searching & S [Count triplet with sum smaller than a given value](#)

Searching & S [merge 2 sorted arrays](#)

Searching & S [print all subarrays with 0 sum](#)

Searching & S [Product array Puzzle](#)

Searching & S [Sort array according to count of set bits](#)

Searching & S [minimum no. of swaps required to sort the array](#)

Searching & S [Bishu and Soldiers](#)

Searching & S [Rasta and Kheshtak](#)

Searching & S [Kth smallest number again](#)

Searching & S [Find pivot element in a sorted array](#)

Searching & S [K-th Element of Two Sorted Arrays](#)

Searching & S [Aggressive cows](#)

Searching & S [Book Allocation Problem](#)

Searching & S [EKOSPOJ:](#)

Searching & S [Job Scheduling Algo](#)

Searching & S [Missing Number in AP](#)

Searching & S [Smallest number with atleastn trailing zeroes infactoria](#)

Searching & S [Painters Partition Problem:](#)

Searching & S [ROTI-Prata SPOJ](#)

Searching & S [DoubleHelix SPOJ](#)

Searching & S [Subset Sums](#)

Searching & S [Findthe inversion count](#)

Searching & S [Implement Merge-sort in-place](#)

Searching & Sorting [Partitioning and Sorting Arrays with Many Repeated Elements](#)

LinkedList [Write a Program to reverse the Linked List. \(Both Iterative and Recursive\)](#)

LinkedList [Reverse a Linked List in group of Given Size. \[Very Imp\]](#)

LinkedList [Write a program to Detect loop in a linked list.](#)

LinkedList [Write a program to Delete loop in a linked list.](#)

LinkedList [Find the starting point of the loop.](#)

LinkedList [Remove Duplicates in a sorted Linked List.](#)

LinkedList [Remove Duplicates in a Un-sorted Linked List.](#)

LinkedList [Write a Program to Move the last element to Front in a](#)

LinkedList [Add "1" to a number represented as a Linked List.](#)

LinkedList [Add two numbers represented by linked lists.](#)

LinkedList [Intersection of two Sorted Linked List.](#)

LinkedList [Intersection Point of two Linked Lists.](#)

LinkedList [Merge Sort For Linked lists.\[Very Important\]](#)

LinkedList [Quicksort for Linked Lists.\[Very Important\]](#)

LinkedList [Find the middle Element of a linked list.](#)

LinkedList [Check if a linked list is a circular linked list.](#)

LinkedList [Split a Circular linked list into two halves.](#)

LinkedList [Write a Program to check whether the Singly Linked list](#)

LinkedList [Deletion from a Circular Linked List.](#)

LinkedList [Reverse a Doubly Linked list.](#)

LinkedList [Find pairs with a given sum in a DLL.](#)

LinkedList [Count triplets in a sorted DLL whose sum is equal to given](#)

LinkedList [Sort a "k"sorted Doubly Linked list.\[Very IMP\]](#)

LinkedList [Rotate DoublyLinked list by N nodes.](#)

LinkedList [Rotate a Doubly Linked list in group of Given Size.\[Very](#)

LinkedList [Can we reverse a linked list in less than O\(n\) ?](#)

LinkedList [Why Quicksort is preferred for. Arrays and Merge Sort](#)

LinkedList [Flatten a Linked List](#)

LinkedList [Sort a LL of 0's, 1's and 2's](#)

LinkedList [Clone a linked list with next and random pointer](#)

Linked Lis [Merge K sorted Linked list](#)
Linked Lis [Multiply 2 no. represented by LL](#)
Linked Lis [Delete nodes which have a greater value on right side](#)
Linked Lis [Segregate even and odd nodes in a Linked List](#)
Linked Lis [Program for n'th node from the end of a Linked List](#)
Linked Lis [Find the first non-repeating character from a stream of](#)

Binary Tre [level order traversal](#)
Binary Tre [Reverse Level Order traversal](#)
Binary Tre [Height of a tree](#)
Binary Tre [Diameter of a tree](#)
Binary Tre [Mirror of a tree](#)
Binary Tre [Inorder Traversal of a tree both using recursion and Iter](#)
Binary Tre [Preorder Traversal of a tree both using recursion and Iter](#)
Binary Tre [Postorder Traversal of a tree both using recursion and Iter](#)
Binary Tre [Left View of a tree](#)
Binary Tre [Right View of Tree](#)
Binary Tre [Top View of a tree](#)
Binary Tre [Bottom View of a tree](#)
Binary Tre [Zig-Zag traversal of a binary tree](#)
Binary Tre [Check if a tree is balanced or not](#)
Binary Tre [Diagnol Traversal of a Binary tree](#)
Binary Tre [Boundary traversal of a Binary tree](#)
Binary Tre [Construct Binary Tree from String with Bracket Represe](#)
Binary Tre [Convert Binary tree into Doubly Linked List](#)
Binary Tre [Convert Binary tree into Sum tree](#)
Binary Tre [Construct Binary tree from Inorder and preorder travers](#)
Binary Tre [Find minimum swaps required to convert a Binary tree i](#)
Binary Tre [Check if Binary tree is Sum tree or not](#)
Binary Tre [Check if all leaf nodes are at same level or not](#)
Binary Tre [Check if a Binary Tree contains duplicate subtrees of siz](#)
Binary Tre [Check if 2 trees are mirror or not](#)

Binary Tree [Sum of Nodes on the Longest path from root to leaf node](#)
Binary Tree [Check if given graph is tree or not. \[IMP \]](#)
Binary Tree [Find Largest subtree sum in a tree](#)
Binary Tree [Maximum Sum of nodes in Binary tree such that no two adjacent nodes have the same value](#)
Binary Tree [Print all "K" Sum paths in a Binary tree](#)
Binary Tree [Find LCA in a Binary tree](#)
Binary Tree [Find distance between 2 nodes in a Binary tree](#)
Binary Tree [Kth Ancestor of node in a Binary tree](#)
Binary Tree [Find all Duplicate subtrees in a Binary tree \[IMP \]](#)
Binary Tree [Tree Isomorphism Problem](#)

Binary Search [Find a value in a BST](#)
Binary Search [Deletion of a node in a BST](#)
Binary Search [Find min and max value in a BST](#)
Binary Search [Find inorder successor and inorder predecessor in a BST](#)
Binary Search [Check if a tree is a BST or not](#)
Binary Search [Populate Inorder successor of all nodes](#)
Binary Search [Find LCA of 2 nodes in a BST](#)
Binary Search [Construct BST from preorder traversal](#)
Binary Search [Convert Binary tree into BST](#)
Binary Search [Convert a normal BST into a Balanced BST](#)
Binary Search [Merge two BST \[V.V.V>IMP \]](#)
Binary Search [Find Kth largest element in a BST](#)
Binary Search [Find Kth smallest element in a BST](#)
Binary Search [Count pairs from 2 BST whose sum is equal to given value](#)
Binary Search [Find the median of BST in O\(n\) time and O\(1\) space](#)
Binary Search [Count BST nodes that lie in a given range](#)
Binary Search [Replace every element with the least greater element on its right](#)
Binary Search [Given "n" appointments, find the conflicting appointments](#)
Binary Search [Check preorder is valid or not](#)
Binary Search [Check whether BST contains Dead end](#)
Binary Search [Largest BST in a Binary Tree \[V.V.V.V.V IMP \]](#)

Binary Search [Flatten BST to sorted list](#)

Greedy [Activity Selection Problem](#)

Greedy [Job Sequencing Problem](#)

Greedy [Huffman Coding](#)

Greedy [Water Connection Problem](#)

Greedy [Fractional Knapsack Problem](#)

Greedy [Greedy Algorithm to find Minimum number of Coins](#)

Greedy [Maximum trains for which stoppage can be provided](#)

Greedy [Minimum Platforms Problem](#)

Greedy [Buy Maximum Stocks if i stocks can be bought on i-th day](#)

Greedy [Find the minimum and maximum amount to buy all N cows](#)

Greedy [Minimize Cash Flow among a given set of friends who have](#)

Greedy [Minimum Cost to cut a board into squares](#)

Greedy [Check if it is possible to survive on Island](#)

Greedy [Find maximum meetings in one room](#)

Greedy [Maximum product subset of an array](#)

Greedy [Maximize array sum after K negations](#)

Greedy [Maximize the sum of \$arr\[i\] * i\$](#)

Greedy [Maximum sum of absolute difference of an array](#)

Greedy [Maximize sum of consecutive differences in a circular array](#)

Greedy [Minimum sum of absolute difference of pairs of two arrays](#)

Greedy [Program for Shortest Job First \(or SJF\) CPU Scheduling](#)

Greedy [Program for Least Recently Used \(LRU\) Page Replacement](#)

Greedy [Smallest subset with sum greater than all other elements](#)

Greedy [Chocolate Distribution Problem](#)

Greedy [DEFKIN -Defense of a Kingdom](#)

Greedy [DIEHARD -DIE HARD](#)

Greedy [GERGOVIA -Wine trading in Gergovia](#)

Greedy [Picking Up Chicks](#)

Greedy [CHOCOLA –Chocolate](#)

Greedy [ARRANGE -Arranging Amplifiers](#)

Greedy [K Centers Problem](#)

Greedy [Minimum Cost of ropes](#)

Greedy [Find smallest number with given number of digits and s](#)

Greedy [Rearrange characters in a string such that no two adjac](#)

Greedy [Find maximum sum possible equal sum of three stacks](#)

BackTracki [Rat in a maze Problem](#)

BackTracki [Printing all solutions in N-Queen Problem](#)

BackTracki [Word Break Problem using Backtracking](#)

BackTracki [Remove Invalid Parentheses](#)

BackTracki [Sudoku Solver](#)

BackTracki [m Coloring Problem](#)

BackTracki [Print all palindromic partitions of a string](#)

BackTracki [Subset Sum Problem](#)

BackTracki [The Knight's tour problem](#)

BackTracki [Tug of War](#)

BackTracki [Find shortest safe route in a path with landmines](#)

BackTracki [Combinational Sum](#)

BackTracki [Find Maximum number possible by doing at-most K swa](#)

BackTracki [Print all permutations of a string](#)

BackTracki [Find if there is a path of more than k length from a sour](#)

BackTracki [Longest Possible Route in a Matrix with Hurdles](#)

BackTracki [Print all possible paths from top left to bottom right of ;](#)

BackTracki [Partition of a set intoK subsets with equal sum](#)

BackTracki [Find the K-th Permutation Sequence of first N natural n](#)

Stacks & Que [Implement Stack from Scratch](#)

Stacks & Que [Implement Queue from Scratch](#)

Stacks & Que [Implement 2 stack in an array](#)

Stacks & Que [find the middle element of a stack](#)

Stacks & Que [Implement "N" stacks in an Array](#)

Stacks & Queue [Check the expression has valid or Balanced parenthesis](#)

Stacks & Queue [Reverse a String using Stack](#)

Stacks & Queue [Design a Stack that supports getMin\(\) in O\(1\) time and \(](#)

Stacks & Queue [Find the next Greater element](#)

Stacks & Queue [The celebrity Problem](#)

Stacks & Queue [Arithmetic Expression evaluation](#)

Stacks & Queue [Evaluation of Postfix expression](#)

Stacks & Queue [Implement a method to insert an element at its bottom](#)

Stacks & Queue [Reverse a stack using recursion](#)

Stacks & Queue [Sort a Stack using recursion](#)

Stacks & Queue [Merge Overlapping Intervals](#)

Stacks & Queue [Largest rectangular Area in Histogram](#)

Stacks & Queue [Length of the Longest Valid Substring](#)

Stacks & Queue [Expression contains redundant bracket or not](#)

Stacks & Queue [Implement Stack using Queue](#)

Stacks & Queue [Implement Stack using Deque](#)

Stacks & Queue [Stack Permutations \(Check if an array is stack permutati](#)

Stacks & Queue [Implement Queue using Stack](#)

Stacks & Queue [Implement "n" queue in an array](#)

Stacks & Queue [Implement a Circular queue](#)

Stacks & Queue [LRU Cache Implementation](#)

Stacks & Queue [Reverse a Queue using recursion](#)

Stacks & Queue [Reverse the first "K" elements of a queue](#)

Stacks & Queue [Interleave the first half of the queue with second half](#)

Stacks & Queue [Find the first circular tour that visits all Petrol Pumps](#)

Stacks & Queue [Minimum time required to rot all oranges](#)

Stacks & Queue [Distance of nearest cell having 1 in a binary matrix](#)

Stacks & Queue [First negative integer in every window of size "k"](#)

Stacks & Queue [Check if all levels of two trees are anagrams or not.](#)

Stacks & Queue [Sum of minimum and maximum elements of all subarra](#)

Stacks & Queue [Minimum sum of squares of character counts in a given](#)

Stacks & Queue [Queue based approach or first non-repeating character](#)

Stacks & Queue [Next Smaller Element](#)

Heap [Implement a Maxheap/MinHeap using arrays and recur](#)
Heap [Sort an Array using heap. \(HeapSort\)](#)
Heap [Maximum of all subarrays of size k.](#)
Heap ["k" largest element in an array](#)
Heap [Kth smallest and largest element in an unsorted array](#)
Heap [Merge "K" sorted arrays. \[IMP \]](#)
Heap [Merge 2 Binary Max Heaps](#)
Heap [Kth largest sum continuous subarrays](#)
Heap [Leetcode- reorganize strings](#)
Heap [Merge "K" Sorted Linked Lists \[V.IMP\]](#)
Heap [Smallest range in "K" Lists](#)
Heap [Median in a stream of Integers](#)
Heap [Check if a Binary Tree is Heap](#)
Heap [Connect "n" ropes with minimum cost](#)
Heap [Convert BST to Min Heap](#)
Heap [Convert min heap to max heap](#)
Heap [Rearrange characters in a string such that no two adjacent](#)
Heap [Minimum sum of two numbers formed from digits of ar](#)

Graph [Create a Graph, print it](#)
Graph [Implement BFS algorithm](#)
Graph [Implement DFS Algo](#)
Graph [Detect Cycle in Directed Graph using BFS/DFS Algo](#)
Graph [Detect Cycle in UnDirected Graph using BFS/DFS Algo](#)
Graph [Search in a Maze](#)
Graph [Minimum Step by Knight](#)
Graph [flood fill algo](#)
Graph [Clone a graph](#)
Graph [Making wired Connections](#)
Graph [word Ladder](#)

Graph [Dijkstra algo](#)
Graph [Implement Topological Sort](#)
Graph [Minimum time taken by each job to be completed given](#)
Graph [Find whether it is possible to finish all tasks or not from](#)
Graph [Find the no. of Islands](#)
Graph [Given a sorted Dictionary of an Alien Language, find ord](#)
Graph [Implement Kruksal's Algorithm](#)
Graph [Implement Prim's Algorithm](#)
Graph [Total no. of Spanning tree in a graph](#)
Graph [Implement Bellman Ford Algorithm](#)
Graph [Implement Floyd warshall Algorithm](#)
Graph [Travelling Salesman Problem](#)
Graph [Graph Colouring Problem](#)
Graph [Snake and Ladders Problem](#)
Graph [Find bridge in a graph](#)
Graph [Count Strongly connected Components\(Kosaraju Algo\)](#)
Graph [Check whether a graph is Bipartite or Not](#)
Graph [Detect Negative cycle in a graph](#)
Graph [Longest path in a Directed Acyclic Graph](#)
Graph [Journey to the Moon](#)
Graph [Cheapest Flights Within K Stops](#)
Graph [Oliver and the Game](#)
Graph [Water Jug problem using BFS](#)
Graph [Water Jug problem using BFS](#)
Graph [Find if there is a path of more than length from a source](#)
Graph [M-Colouring Problem](#)
Graph [Minimum edges to reverse to make path from source to](#)
Graph [Paths to travel each node using each edge\(Seven Bridges\)](#)
Graph [Vertex Cover Problem](#)
Graph [Chinese Postman or Route Inspection](#)
Graph [Number of Triangles in a Directed and Undirected Graph](#)
Graph [Minimise the cashflow among a given set of friends who](#)
Graph [Two Clique Problem](#)

Trie [Construct a trie from scratch](#)
Trie [Find shortest unique prefix for every word in a given list](#)
Trie [Word Break Problem | \(Trie solution\)](#)
Trie [Given a sequence of words, print all anagrams together](#)
Trie [Implement a Phone Directory](#)
Trie [Print unique rows in a given boolean matrix](#)

Dynamic Program [Coin Change Problem](#)
Dynamic Program [Knapsack Problem](#)
Dynamic Program [Binomial Coefficient Problem](#)
Dynamic Program [Permutation Coefficient Problem](#)
Dynamic Program [Program for nth Catalan Number](#)
Dynamic Program [Matrix Chain Multiplication](#)
Dynamic Program [Edit Distance](#)
Dynamic Program [Subset Sum Problem](#)
Dynamic Program [Friends Pairing Problem](#)
Dynamic Program [Gold Mine Problem](#)
Dynamic Program [Assembly Line Scheduling Problem](#)
Dynamic Program [Painting the Fence problem](#)
Dynamic Program [Maximize The Cut Segments](#)
Dynamic Program [Longest Common Subsequence](#)
Dynamic Program [Longest Repeated Subsequence](#)
Dynamic Program [Longest Increasing Subsequence](#)
Dynamic Program [Space Optimized Solution of LCS](#)
Dynamic Program [LCS \(Longest Common Subsequence\) of three strings](#)
Dynamic Program [Maximum Sum Increasing Subsequence](#)
Dynamic Program [Count all subsequences having product less than K](#)
Dynamic Program [Longest subsequence such that difference between adjacent elements is at least K](#)
Dynamic Program [Maximum subsequence sum such that no three are consecutive](#)
Dynamic Program [Egg Dropping Problem](#)

Dynamic Program [Maximum Length Chain of Pairs](#)

Dynamic Program [Maximum size square sub-matrix with all 1s](#)

Dynamic Program [Maximum sum of pairs with specific difference](#)

Dynamic Program [Min Cost Path Problem](#)

Dynamic Program [Maximum difference of zeros and ones in binary string](#)

Dynamic Program [Minimum number of jumps to reach end](#)

Dynamic Program [Minimum cost to fill given weight in a bag](#)

Dynamic Program [Minimum removals from array to make max – min \$\leq K\$](#)

Dynamic Program [Longest Common Substring](#)

Dynamic Program [Count number of ways to reach a given score in a game](#)

Dynamic Program [Count Balanced Binary Trees of Height h](#)

Dynamic Program [Largest Sum Contiguous Subarray \[V>V>V>V IMP\]](#)

Dynamic Program [Smallest sum contiguous subarray](#)

Dynamic Program [Unbounded Knapsack \(Repetition of items allowed\)](#)

Dynamic Program [Word Break Problem](#)

Dynamic Program [Largest Independent Set Problem](#)

Dynamic Program [Partition problem](#)

Dynamic Program [Longest Palindromic Subsequence](#)

Dynamic Program [Count All Palindromic Subsequence in a given String](#)

Dynamic Program [Longest Palindromic Substring](#)

Dynamic Program [Longest alternating subsequence](#)

Dynamic Program [Weighted Job Scheduling](#)

Dynamic Program [Coin game winner where every player has three choices](#)

Dynamic Program [Count Derangements \(Permutation such that no element is in its original position\)](#)

Dynamic Program [Maximum profit by buying and selling a share at most twice](#)

Dynamic Program [Optimal Strategy for a Game](#)

Dynamic Program [Optimal Binary Search Tree](#)

Dynamic Program [Palindrome Partitioning Problem](#)

Dynamic Program [Word Wrap Problem](#)

Dynamic Program [Mobile Numeric Keypad Problem \[IMP\]](#)

Dynamic Program [Boolean Parenthesization Problem](#)

Dynamic Program [Largest rectangular sub-matrix whose sum is 0](#)

Dynamic Program [Largest area rectangular sub-matrix with equal number of 0s and 1s](#)

Dynamic Program [Maximum sum rectangle in a 2D matrix](#)

Dynamic Program [Maximum profit by buying and selling a share at most k](#)

Dynamic Program [Find if a string is interleaved of two other strings](#)

Dynamic Program [Maximum Length of Pair Chain](#)

Bit Manipulation [Count set bits in an integer](#)

Bit Manipulation [Find the two non-repeating elements in an array of repeating elements](#)

Bit Manipulation [Count number of bits to be flipped to convert A to B](#)

Bit Manipulation [Count total set bits in all numbers from 1 to n](#)

Bit Manipulation [Program to find whether a no is power of two](#)

Bit Manipulation [Find position of the only set bit](#)

Bit Manipulation [Copy set bits in a range](#)

Bit Manipulation [Divide two integers without using multiplication, division and modulo operator](#)

Bit Manipulation [Calculate square of a number without using *, / and pow](#)

Bit Manipulation [Power Set](#)

It appears in its original position) [IMPORTANT]

