None
Questions by Love Babbar:
None

None
Youtube Channel: https://www.youtube.com/channel/UCQHLxxBFrbfdrk1jF0moTpw
None

None
None
None

Topic:
Problem:
Done [yes or no]

None
None
<->

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 without using any sorting algo
<->

Array
Move all the negative elements to one side of the array
<->

Array
Find the Union and Intersection of the two sorted arrays.
<->

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 number

<->

Array

find common elements In 3 sorted arrays

<->

Array

Rearrange the array in alternating positive and negative items with O(1) extra space

<->

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 coinsecutive subsequence

<->

Array

Given an array of size n and a number k, fin all elements that appear more than " n/k " times.

<->

Array

Maximum profit by buying and selling a share atmost 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 together
<->

Array
Minimum no. of operations required to make an array palindrome
<->

Array
Median of 2 sorted arrays of equal size
<->

Array
Median of 2 sorted arrays of different size
<->

None
None
<->

None
None
<->

Matrix
Spiral traversal on a Matrix
<->

Matrix

Search an element in a matriix

<->

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 sorted matrix

<->

Matrix

Maximum size rectangle

<->

Matrix

Find a specific pair in matrix

<->

Matrix

Rotate matrix by 90 degrees

<->

Matrix

Kth smallest element in a row-cpumn wise sorted matrix

<->

Matrix

Common elements in all rows of a given matrix

<->

None

None

None

None

None

None

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 another

<->

String

Write a Program to check whether a string is a valid shuffle of two strings or not

<->

String

Count and Say problem

<->

String

Write a program to find the longest Palindrome in a string.[Longest palindromic Substring]

<->

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 1’s
<->

String
Word Wrap Problem [VERY IMP].
<->

String
EDIT Distance [Very Imp]
<->

String
Find next greater number with same set of digits. [Very Very IMP]
<->

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 keypad sequence.
<->

String
Minimum number of bracket reversals needed to make an expression balanced.
<->

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 strings.

<->

String

Program to generate all possible valid IP addresses from given string.

<->

String

Write a program tofind the smallest window that contains all characters of string itself.

<->

String
Rearrange characters in a string such that no two adjacent are same
<->

String
Minimum characters to be added at front to make string palindrome
<->

String
Given a sequence of words, print all anagrams together
<->

String
Find the smallest window in a string containing all characters of another string
<->

String
Recursively remove all adjacent duplicates
<->

String
String matching where one string contains wildcard characters
<->

String
Function to find Number of customers who could not get a computer
<->

String
Transform One String to Another using Minimum Number of Given Operation
<->

String
Check if two given strings are isomorphic to each other
<->

String
Recursively print all sentences that can be formed from list of word lists
<->

None
None
None

None

None

None

Searching & Sorting

Find first and last positions of an element in a sorted array

<->

Searching & Sorting

Find a Fixed Point (Value equal to index) in a given array

<->

Searching & Sorting

Search in a rotated sorted array

<->

Searching & Sorting

square root of an integer

<->

Searching & Sorting

Maximum and minimum of an array using minimum number of comparisons

<->

Searching & Sorting

Optimum location of point to minimize total distance

<->

Searching & Sorting

Find the repeating and the missing

<->

Searching & Sorting

find majority element

<->

Searching & Sorting

Searching in an array where adjacent differ by at most k

<->

Searching & Sorting

find a pair with a given difference

<->

Searching & Sorting

find four elements that sum to a given value

<->

Searching & Sorting

maximum sum such that no 2 elements are adjacent

<->

Searching & Sorting

Count triplet with sum smaller than a given value

<->

Searching & Sorting

merge 2 sorted arrays

<->

Searching & Sorting

print all subarrays with 0 sum

<->

Searching & Sorting

Product array Puzzle

<->

Searching & Sorting

Sort array according to count of set bits

<->

Searching & Sorting

minimum no. of swaps required to sort the array

<->

Searching & Sorting

Bishu and Soldiers

<->

Searching & Sorting

Rasta and Kheshtak

<->

Searching & Sorting

Kth smallest number again

<->

Searching & Sorting

Find pivot element in a sorted array

<->

Searching & Sorting

K-th Element of Two Sorted Arrays

<->

Searching & Sorting

Aggressive cows

<->

Searching & Sorting

Book Allocation Problem

<->

Searching & Sorting

EKOSPOJ:

<->

Searching & Sorting

Job Scheduling Algo

<->

Searching & Sorting

Missing Number in AP

<->

Searching & Sorting

Smallest number with atleastn trailing zeroes infactorial

<->

Searching & Sorting

Painters Partition Problem:

<->

Searching & Sorting

ROTI-Prata SPOJ

<->

Searching & Sorting

DoubleHelix SPOJ

<->

Searching & Sorting

Subset Sums

<->

Searching & Sorting

Findthe inversion count

<->

Searching & Sorting

Implement Merge-sort in-place

<->

Searching & Sorting

Partitioning and Sorting Arrays with Many Repeated Entries

<->

None

None

None

None

None

None

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 Linked List.

<->

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 is a palindrome or not.

<->

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 value “X”.

<->

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 IMP]

<->

LinkedList

Can we reverse a linked list in less than O(n) ?

<->

LinkedList

Why Quicksort is preferred for. Arrays and Merge Sort for LinkedLists ?

<->

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

<->

LinkedList

Merge K sorted Linked list

<->

LinkedList

Multiply 2 no. represented by LL

<->

LinkedList

Delete nodes which have a greater value on right side

<->

LinkedList

Segregate even and odd nodes in a Linked List

<->

LinkedList

Program for n’th node from the end of a Linked List

<->

LinkedList

Find the first non-repeating character from a stream of characters

<->

None

None

None

None

None

None

Binary Trees
level order traversal
<->

Binary Trees
Reverse Level Order traversal
<->

Binary Trees
Height of a tree
<->

Binary Trees
Diameter of a tree
<->

Binary Trees
Mirror of a tree
<->

Binary Trees
Inorder Traversal of a tree both using recursion and Iteration
<->

Binary Trees
Preorder Traversal of a tree both using recursion and Iteration
<->

Binary Trees
Postorder Traversal of a tree both using recursion and Iteration
<->

Binary Trees
Left View of a tree
<->

Binary Trees
Right View of Tree
<->

Binary Trees
Top View of a tree
<->

Binary Trees

Bottom View of a tree

<->

Binary Trees

Zig-Zag traversal of a binary tree

<->

Binary Trees

Check if a tree is balanced or not

<->

Binary Trees

Diagnol Traversal of a Binary tree

<->

Binary Trees

Boundary traversal of a Binary tree

<->

Binary Trees

Construct Binary Tree from String with Bracket Representation

<->

Binary Trees

Convert Binary tree into Doubly Linked List

<->

Binary Trees

Convert Binary tree into Sum tree

<->

Binary Trees

Construct Binary tree from Inorder and preorder traversal

<->

Binary Trees

Find minimum swaps required to convert a Binary tree into BST

<->

Binary Trees

Check if Binary tree is Sum tree or not

<->

Binary Trees

Check if all leaf nodes are at same level or not

<->

Binary Trees

Check if a Binary Tree contains duplicate subtrees of size 2 or more [IMP]

<->

Binary Trees

Check if 2 trees are mirror or not

<->

Binary Trees

Sum of Nodes on the Longest path from root to leaf node

<->

Binary Trees

Check if given graph is tree or not. [IMP]

<->

Binary Trees

Find Largest subtree sum in a tree

<->

Binary Trees

Maximum Sum of nodes in Binary tree such that no two are adjacent

<->

Binary Trees

Print all "K" Sum paths in a Binary tree

<->

Binary Trees

Find LCA in a Binary tree

<->

Binary Trees

Find distance between 2 nodes in a Binary tree

<->

Binary Trees

Kth Ancestor of node in a Binary tree

<->

Binary Trees

Find all Duplicate subtrees in a Binary tree [IMP]

<->

Binary Trees

Tree Isomorphism Problem

<->

None

None

None

None

None

None

Binary Search Trees

Fina a value in a BST

<->

Binary Search Trees

Deletion of a node in a BST

<->

Binary Search Trees

Find min and max value in a BST

<->

Binary Search Trees

Find inorder successor and inorder predecessor in a BST

<->

Binary Search Trees

Check if a tree is a BST or not

<->

Binary Search Trees

Populate Inorder successor of all nodes

<->

Binary Search Trees

Find LCA of 2 nodes in a BST

<->

Binary Search Trees

Construct BST from preorder traversal

<->

Binary Search Trees

Convert Binary tree into BST

<->

Binary Search Trees

Convert a normal BST into a Balanced BST

<->

Binary Search Trees

Merge two BST [V.V.V>IMP]

<->

Binary Search Trees

Find Kth largest element in a BST

<->

Binary Search Trees

Find Kth smallest element in a BST

<->

Binary Search Trees

Count pairs from 2 BST whose sum is equal to given value "X"

<->

Binary Search Trees

Find the median of BST in O(n) time and O(1) space

<->

Binary Search Trees

Count BST ndoes that lie in a given range

<->

Binary Search Trees

Replace every element with the least greater element on its right

<->

Binary Search Trees

Given "n" appointments, find the conflicting appointments

<->

Binary Search Trees
Check preorder is valid or not
<->

Binary Search Trees
Check whether BST contains Dead end
<->

Binary Search Trees
Largest BST in a Binary Tree [V.V.V.V.V IMP]
<->

Binary Search Trees
Flatten BST to sorted list
<->

None
None
None

None
None
None
None

Greedy
Activity Selection Problem
<->

Greedy
Job SequencingProblem
<->

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 candies

<->

Greedy

Minimize Cash Flow among a given set of friends who have borrowed money from each other

<->

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 algorithm

<->

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 sum of digits
<->

Greedy
Rearrange characters in a string such that no two adjacent are same
<->

Greedy
Find maximum sum possible equal sum of three stacks
<->

None
None
None

None
None
None

BackTracking
Rat in a maze Problem
<->

BackTracking

Printing all solutions in N-Queen Problem

<->

BackTracking

Word Break Problem using Backtracking

<->

BackTracking

Remove Invalid Parentheses

<->

BackTracking

Sudoku Solver

<->

BackTracking

m Coloring Problem

<->

BackTracking

Print all palindromic partitions of a string

<->

BackTracking

Subset Sum Problem

<->

BackTracking

The Knight’s tour problem

<->

BackTracking

Tug of War

<->

BackTracking

Find shortest safe route in a path with landmines

<->

BackTracking

Combinational Sum

<->

BackTracking

Find Maximum number possible by doing at-most K swaps

<->

BackTracking

Print all permutations of a string

<->

BackTracking

Find if there is a path of more than k length from a source

<->

BackTracking

Longest Possible Route in a Matrix with Hurdles

<->

BackTracking

Print all possible paths from top left to bottom right of a mXn matrix

<->

BackTracking

Partition of a set intoK subsets with equal sum

<->

BackTracking

Find the K-th Permutation Sequence of first N natural numbers

<->

None

None

None

None

None

None

Stacks & Queues

Implement Stack from Scratch

<->

Stacks & Queues

Implement Queue from Scratch

<->

Stacks & Queues

Implement 2 stack in an array

<->

Stacks & Queues

find the middle element of a stack

<->

Stacks & Queues

Implement "N" stacks in an Array

<->

Stacks & Queues

Check the expression has valid or Balanced parenthesis or not.

<->

Stacks & Queues

Reverse a String using Stack

<->

Stacks & Queues

Design a Stack that supports getMin() in O(1) time and O(1) extra space.

<->

Stacks & Queues

Find the next Greater element

<->

Stacks & Queues

The celebrity Problem

<->

Stacks & Queues

Arithmetic Expression evaluation

<->

Stacks & Queues

Evaluation of Postfix expression

<->

Stacks & Queues

Implement a method to insert an element at its bottom without using any other data structure.

<->

Stacks & Queues

Reverse a stack using recursion

<->

Stacks & Queues

Sort a Stack using recursion

<->

Stacks & Queues

Merge Overlapping Intervals

<->

Stacks & Queues

Largest rectangular Area in Histogram

<->

Stacks & Queues

Length of the Longest Valid Substring

<->

Stacks & Queues

Expression contains redundant bracket or not

<->

Stacks & Queues

Implement Stack using Queue

<->

Stacks & Queues

Implement Stack using Deque

<->

Stacks & Queues

Stack Permutations (Check if an array is stack permutation of other)

<->

Stacks & Queues

Implement Queue using Stack

<->

Stacks & Queues

Implement "n" queue in an array

<->

Stacks & Queues

Implement a Circular queue

<->

Stacks & Queues

LRU Cache Implementationa

<->

Stacks & Queues

Reverse a Queue using recursion

<->

Stacks & Queues

Reverse the first “K” elements of a queue

<->

Stacks & Queues

Interleave the first half of the queue with second half

<->

Stacks & Queues

Find the first circular tour that visits all Petrol Pumps

<->

Stacks & Queues

Minimum time required to rot all oranges

<->

Stacks & Queues

Distance of nearest cell having 1 in a binary matrix

<->

Stacks & Queues

First negative integer in every window of size “k”

<->

Stacks & Queues

Check if all levels of two trees are anagrams or not.

<->

Stacks & Queues

Sum of minimum and maximum elements of all subarrays of size “k”.

<->

Stacks & Queues

Minimum sum of squares of character counts in a given string after removing “k” characters.

<->

Stacks & Queues

Queue based approach or first non-repeating character in a stream.

<->

Stacks & Queues

Next Smaller Element

<->

None

None

None

None

None

None

None

Heap

Implement a Maxheap/MinHeap using arrays and recursion.

<->

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 are same.

<->

Heap

Minimum sum of two numbers formed from digits of an array

<->

None

None

None

None

None

None

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 by a Directed Acyclic Graph

<->

Graph

Find whether it is possible to finish all tasks or not from given dependencies

<->

Graph

Find the no. of Isalnds

<->

Graph

Given a sorted Dictionary of an Alien Language, find order of characters

<->

Graph

Implement Kruksal’sAlgorithm

<->

Graph

Implement Prim’s Algorithm

<->

Graph

Total no. of Spanning tree in a graph

<->

Graph

Implement Bellman Ford Algorithm

<->

Graph

Implement Floyd warshallAlgorithm

<->

Graph

Travelling Salesman Problem

<->

Graph

Graph ColouringProblem

<->

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 thank length from a source

<->

Graph

M-ColouringProblem

<->

Graph

Minimum edges to reverse o make path from source to destination

<->

Graph

Paths to travel each nodes 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 have borrowed money from each other

<->

Graph

Two Clique Problem

<->

None

None

None

None

None

None

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

<->

None
None
None

None
None
None

Dynamic Programming
Coin ChangeProblem
<->

Dynamic Programming
Knapsack Problem
<->

Dynamic Programming
Binomial CoefficientProblem
<->

Dynamic Programming
Permutation CoefficientProblem
<->

Dynamic Programming
Program for nth Catalan Number
<->

Dynamic Programming
Matrix Chain Multiplication
<->

Dynamic Programming
Edit Distance
<->

Dynamic Programming
Subset Sum Problem
<->

Dynamic Programming
Friends Pairing Problem
<->

Dynamic Programming
Gold Mine Problem
<->

Dynamic Programming
Assembly Line SchedulingProblem
<->

Dynamic Programming
Painting the Fenceproblem
<->

Dynamic Programming
Maximize The Cut Segments
<->

Dynamic Programming
Longest Common Subsequence
<->

Dynamic Programming
Longest Repeated Subsequence
<->

Dynamic Programming
Longest Increasing Subsequence
<->

Dynamic Programming
Space Optimized Solution of LCS
<->

Dynamic Programming
LCS (Longest Common Subsequence) of three strings
<->

Dynamic Programming
Maximum Sum Increasing Subsequence
<->

Dynamic Programming
Count all subsequences having product less than K
<->

Dynamic Programming

Longest subsequence such that difference between adjacent is one

<->

Dynamic Programming

Maximum subsequence sum such that no three are consecutive

<->

Dynamic Programming

Egg Dropping Problem

<->

Dynamic Programming

Maximum Length Chain of Pairs

<->

Dynamic Programming

Maximum size square sub-matrix with all 1s

<->

Dynamic Programming

Maximum sum of pairs with specific difference

<->

Dynamic Programming

Min Cost PathProblem

<->

Dynamic Programming

Maximum difference of zeros and ones in binary string

<->

Dynamic Programming

Minimum number of jumps to reach end

<->

Dynamic Programming

Minimum cost to fill given weight in a bag

<->

Dynamic Programming

Minimum removals from array to make max –min <= K

<->

Dynamic Programming
Longest Common Substring
<->

Dynamic Programming
Count number of ways to reacha given score in a game
<->

Dynamic Programming
Count Balanced Binary Trees of Height h
<->

Dynamic Programming
LargestSum Contiguous Subarray [V>V>V>V IMP]
<->

Dynamic Programming
Smallest sum contiguous subarray
<->

Dynamic Programming
Unbounded Knapsack (Repetition of items allowed)
<->

Dynamic Programming
Word Break Problem
<->

Dynamic Programming
Largest Independent Set Problem
<->

Dynamic Programming
Partition problem
<->

Dynamic Programming
Longest Palindromic Subsequence
<->

Dynamic Programming
Count All Palindromic Subsequence in a given String
<->

Dynamic Programming
Longest Palindromic Substring
<->

Dynamic Programming
Longest alternating subsequence
<->

Dynamic Programming
Weighted Job Scheduling
<->

Dynamic Programming
Coin game winner where every player has three choices
<->

Dynamic Programming
Count Derangements (Permutation such that no element appears in its original position) [IMPORTANT]
<->

Dynamic Programming
Maximum profit by buying and selling a share at most twice [IMP]
<->

Dynamic Programming
Optimal Strategy for a Game
<->

Dynamic Programming
Optimal Binary Search Tree
<->

Dynamic Programming
Palindrome PartitioningProblem
<->

Dynamic Programming
Word Wrap Problem
<->

Dynamic Programming
Mobile Numeric Keypad Problem [IMP]
<->

Dynamic Programming
Boolean Parenthesization Problem
<->

Dynamic Programming
Largest rectangular sub-matrix whose sum is 0
<->

Dynamic Programming
Largest area rectangular sub-matrix with equal number of 1’s and 0’s [IMP]
<->

Dynamic Programming
Maximum sum rectangle in a 2D matrix
<->

Dynamic Programming
Maximum profit by buying and selling a share at most k times
<->

Dynamic Programming
Find if a string is interleaved of two other strings
<->

Dynamic Programming
Maximum Length of Pair Chain
<->

None
None
None

None
None
None

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 mod operator

<->

Bit Manipulation

Calculate square of a number without using *, / and pow()

<->

Bit Manipulation

Power Set

<->

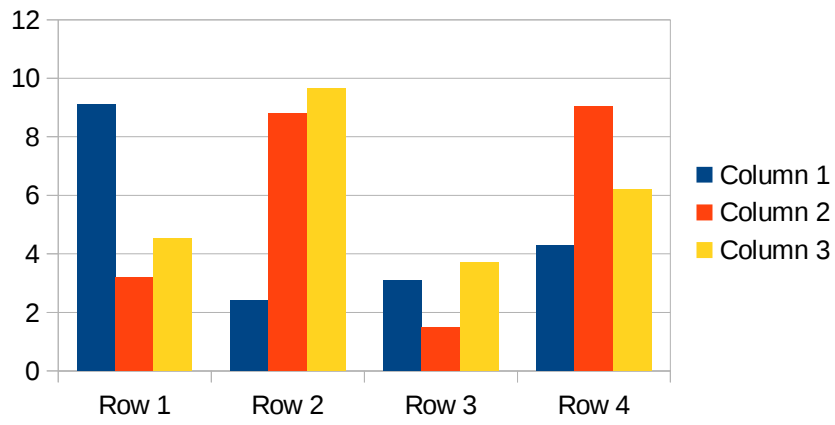
Lorem ipsum

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc ac faucibus odio.

Vestibulum neque massa, scelerisque sit amet ligula eu, congue molestie mi. Praesent ut varius sem. Nullam at porttitor arcu, nec lacinia nisi. Ut ac dolor vitae odio interdum condimentum. **Vivamus dapibus sodales ex, vitae malesuada ipsum cursus convallis. Maecenas sed egestas nulla, ac condimentum orci.** Mauris diam felis, vulputate ac suscipit et, iaculis non est. Curabitur semper arcu ac ligula semper, nec luctus nisl blandit. Integer lacinia ante ac libero lobortis imperdiet. *Nullam mollis convallis ipsum, ac accumsan nunc vehicula vitae.* Nulla eget justo in felis tristique fringilla. Morbi sit amet tortor quis risus auctor condimentum. Morbi in ullamcorper elit. Nulla iaculis tellus sit amet mauris tempus fringilla.

Maecenas mauris lectus, lobortis et purus mattis, blandit dictum tellus.

- **Maecenas non lorem quis tellus placerat varius.**
- *Nulla facilisi.*
- Aenean congue fringilla justo ut aliquam.
- Mauris id ex erat. Nunc vulputate neque vitae justo facilisis, non condimentum ante sagittis.
- Morbi viverra semper lorem nec molestie.
- Maecenas tincidunt est efficitur ligula euismod, sit amet ornare est vulputate.



In non mauris justo. Duis vehicula mi vel mi pretium, a viverra erat efficitur. Cras aliquam est ac eros varius, id iaculis dui auctor. Duis pretium neque ligula, et pulvinar mi placerat et. Nulla nec nunc sit amet nunc posuere vestibulum. Ut id neque eget tortor mattis tristique. Donec ante est, blandit sit amet tristique vel, lacinia pulvinar arcu. Pellentesque scelerisque fermentum erat, id posuere justo pulvinar ut. Cras id eros sed enim aliquam lobortis. Sed lobortis nisl ut eros efficitur tincidunt. Cras justo mi, porttitor quis mattis vel, ultricies ut purus. Ut facilisis et lacus eu cursus.

In eleifend velit vitae libero sollicitudin euismod. Fusce vitae vestibulum velit. Pellentesque vulputate lectus quis pellentesque commodo. Aliquam erat volutpat. Vestibulum in egestas velit. Pellentesque fermentum nisl vitae fringilla venenatis. Etiam id mauris vitae orci maximus ultricies.

Cras fringilla ipsum magna, in fringilla dui commodo a.

	Lorem ipsum	Lorem ipsum	Lorem ipsum
1	In eleifend velit vitae libero sollicitudin euismod.	Lorem	
2	Cras fringilla ipsum magna, in fringilla dui commodo a.	Ipsum	
3	Aliquam erat volutpat.	Lorem	
4	Fusce vitae vestibulum velit.	Lorem	
5	Etiam vehicula luctus fermentum.	Ipsum	

Etiam vehicula luctus fermentum. In vel metus congue, pulvinar lectus vel, fermentum dui. Maecenas ante orci, egestas ut aliquet sit amet, sagittis a magna. Aliquam ante quam, pellentesque ut dignissim quis, laoreet eget est. Aliquam erat volutpat. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Ut ullamcorper justo sapien, in cursus libero viverra eget. Vivamus auctor imperdiet urna, at pulvinar leo posuere laoreet. Suspendisse neque nisl, fringilla at iaculis scelerisque, ornare vel dolor. Ut et pulvinar nunc. Pellentesque fringilla mollis efficitur. Nullam venenatis commodo imperdiet. Morbi velit neque, semper quis lorem quis, efficitur dignissim ipsum. Ut ac lorem sed turpis imperdiet eleifend sit amet id sapien.

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Nunc ac faucibus odio. Vestibulum neque massa, scelerisque sit amet ligula eu, congue molestie mi. Praesent ut varius sem. Nullam at porttitor arcu, nec lacinia nisi. Ut ac dolor vitae odio interdum condimentum. Vivamus dapibus sodales ex, vitae malesuada ipsum cursus convallis. Maecenas sed egestas nulla, ac condimentum orci. Mauris diam felis, vulputate ac suscipit et, iaculis non est. Curabitur semper arcu ac ligula semper, nec luctus nisl blandit. Integer lacinia ante ac libero lobortis imperdiet. Nullam mollis convallis ipsum, ac accumsan nunc vehicula vitae. Nulla eget justo in felis tristique fringilla. Morbi sit amet tortor quis risus auctor condimentum. Morbi in ullamcorper elit. Nulla iaculis tellus sit amet mauris tempus fringilla.

Maecenas mauris lectus, lobortis et purus mattis, blandit dictum tellus.

Maecenas non lorem quis tellus placerat varius. Nulla facilisi. Aenean congue fringilla justo ut aliquam. Mauris id ex erat. Nunc vulputate neque vitae justo facilisis, non condimentum ante sagittis. Morbi viverra semper lorem nec molestie. Maecenas tincidunt est efficitur ligula euismod, sit amet ornare est vulputate.

In non mauris justo. Duis vehicula mi vel mi pretium, a viverra erat efficitur. Cras aliquam est ac eros varius, id iaculis dui auctor. Duis pretium neque ligula, et pulvinar mi placerat et. Nulla nec nunc sit amet nunc posuere vestibulum. Ut id neque eget tortor mattis tristique. Donec ante est, blandit sit amet tristique vel, lacinia pulvinar arcu. Pellentesque scelerisque fermentum erat, id posuere justo pulvinar ut. Cras id eros sed enim aliquam lobortis. Sed lobortis nisl ut eros efficitur tincidunt. Cras justo mi, porttitor quis mattis vel, ultricies ut purus. Ut facilisis et lacus eu cursus.

In eleifend velit vitae libero sollicitudin euismod.

Fusce vitae vestibulum velit. Pellentesque vulputate lectus quis pellentesque commodo. Aliquam erat volutpat. Vestibulum in egestas velit. Pellentesque fermentum nisl vitae fringilla venenatis. Etiam id mauris vitae orci maximus ultricies. Cras fringilla ipsum magna, in fringilla dui commodo a.

Etiam vehicula luctus fermentum. In vel metus congue, pulvinar lectus vel, fermentum dui. Maecenas ante orci, egestas ut aliquet sit amet, sagittis a magna. Aliquam ante quam, pellentesque ut dignissim quis, laoreet eget est. Aliquam erat volutpat. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Ut ullamcorper justo sapien, in cursus libero viverra eget. Vivamus auctor imperdiet urna, at pulvinar leo posuere laoreet. Suspendisse neque nisl, fringilla at iaculis scelerisque, ornare vel dolor. Ut et pulvinar nunc. Pellentesque fringilla mollis efficitur. Nullam venenatis commodo imperdiet. Morbi velit neque, semper quis lorem quis, efficitur dignissim ipsum. Ut ac lorem sed turpis imperdiet eleifend sit amet id sapien.

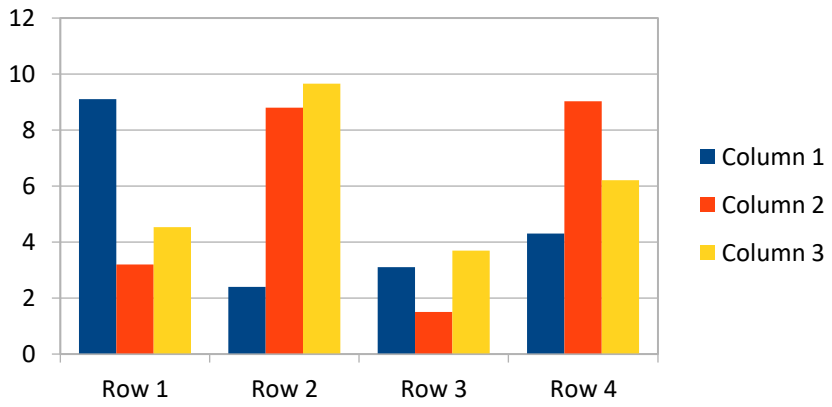
Lorem ipsum

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc ac faucibus odio.

Vestibulum neque massa, scelerisque sit amet ligula eu, congue molestie mi. Praesent ut varius sem. Nullam at porttitor arcu, nec lacinia nisi. Ut ac dolor vitae odio interdum condimentum. **Vivamus dapibus sodales ex, vitae malesuada ipsum cursus convallis. Maecenas sed egestas nulla, ac condimentum orci.** Mauris diam felis, vulputate ac suscipit et, iaculis non est. Curabitur semper arcu ac ligula semper, nec luctus nisl blandit. Integer lacinia ante ac libero lobortis imperdiet. *Nullam mollis convallis ipsum, ac accumsan nunc vehicula vitae.* Nulla eget justo in felis tristique fringilla. Morbi sit amet tortor quis risus auctor condimentum. Morbi in ullamcorper elit. Nulla iaculis tellus sit amet mauris tempus fringilla.

Maecenas mauris lectus, lobortis et purus mattis, blandit dictum tellus.

- **Maecenas non lorem quis tellus placerat varius.**
- *Nulla facilisi.*
- Aenean congue fringilla justo ut aliquam.
- Mauris id ex erat. Nunc vulputate neque vitae justo facilisis, non condimentum ante sagittis.
- Morbi viverra semper lorem nec molestie.
- Maecenas tincidunt est efficitur ligula euismod, sit amet ornare est vulputate.



In non mauris justo. Duis vehicula mi vel mi pretium, a viverra erat efficitur. Cras aliquam est ac eros varius, id iaculis dui auctor. Duis pretium neque ligula, et pulvinar mi placerat et. Nulla nec nunc sit amet nunc posuere vestibulum. Ut id neque eget tortor mattis tristique. Donec ante est, blandit sit amet tristique vel, lacinia pulvinar arcu. Pellentesque scelerisque fermentum erat, id posuere justo pulvinar ut. Cras id eros sed enim aliquam lobortis. Sed lobortis nisl ut eros efficitur tincidunt. Cras justo mi, porttitor quis mattis vel, ultricies ut purus. Ut facilisis et lacus eu cursus.

In eleifend velit vitae libero sollicitudin euismod. Fusce vitae vestibulum velit. Pellentesque vulputate lectus quis pellentesque commodo. Aliquam erat volutpat. Vestibulum in egestas velit. Pellentesque fermentum nisl vitae fringilla venenatis. Etiam id mauris vitae orci maximus ultricies.

Cras fringilla ipsum magna, in fringilla dui commodo a.

	Lorem ipsum	Lorem ipsum	Lorem ipsum
1	In eleifend velit vitae libero sollicitudin euismod.	Lorem	
2	Cras fringilla ipsum magna, in fringilla dui commodo a.	Ipsum	
3	Aliquam erat volutpat.	Lorem	
4	Fusce vitae vestibulum velit.	Lorem	
5	Etiam vehicula luctus fermentum.	Ipsum	

Etiam vehicula luctus fermentum. In vel metus congue, pulvinar lectus vel, fermentum dui. Maecenas ante orci, egestas ut aliquet sit amet, sagittis a magna. Aliquam ante quam, pellentesque ut dignissim quis, laoreet eget est. Aliquam erat volutpat. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Ut

ullamcorper justo sapien, in cursus libero viverra eget. Vivamus auctor imperdiet urna, at pulvinar leo posuere laoreet. Suspendisse neque nisl, fringilla at iaculis scelerisque, ornare vel dolor. Ut et pulvinar nunc. Pellentesque fringilla mollis efficitur. Nullam venenatis commodo imperdiet. Morbi velit neque, semper quis lorem quis, efficitur dignissim ipsum. Ut ac lorem sed turpis imperdiet eleifend sit amet id sapien.

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Nunc ac faucibus odio. Vestibulum neque massa, scelerisque sit amet ligula eu, congue molestie mi. Praesent ut varius sem. Nullam at porttitor arcu, nec lacinia nisi. Ut ac dolor vitae odio interdum condimentum. Vivamus dapibus sodales ex, vitae malesuada ipsum cursus convallis. Maecenas sed egestas nulla, ac condimentum orci. Mauris diam felis, vulputate ac suscipit et, iaculis non est. Curabitur semper arcu ac ligula semper, nec luctus nisl blandit. Integer lacinia ante ac libero lobortis imperdiet. Nullam mollis convallis ipsum, ac accumsan nunc vehicula vitae. Nulla eget justo in felis tristique fringilla. Morbi sit amet tortor quis risus auctor condimentum. Morbi in ullamcorper elit. Nulla iaculis tellus sit amet mauris tempus fringilla.

Maecenas mauris lectus, lobortis et purus mattis, blandit dictum tellus.

Maecenas non lorem quis tellus placerat varius. Nulla facilisi. Aenean congue fringilla justo ut aliquam. Mauris id ex erat. Nunc vulputate neque vitae justo facilisis, non condimentum ante sagittis. Morbi viverra semper lorem nec molestie. Maecenas tincidunt est efficitur ligula euismod, sit amet ornare est vulputate.

In non mauris justo. Duis vehicula mi vel mi pretium, a viverra erat efficitur. Cras aliquam est ac eros varius, id iaculis dui auctor. Duis pretium neque ligula, et pulvinar mi placerat et. Nulla nec nunc sit amet nunc posuere vestibulum. Ut id neque eget tortor mattis tristique. Donec ante est, blandit sit amet tristique vel, lacinia pulvinar arcu. Pellentesque scelerisque fermentum erat, id posuere justo pulvinar ut. Cras id eros sed enim aliquam lobortis. Sed lobortis nisl ut eros efficitur tincidunt. Cras justo mi, porttitor quis mattis vel, ultricies ut purus. Ut facilisis et lacus eu cursus.

In eleifend velit vitae libero sollicitudin euismod.

Fusce vitae vestibulum velit. Pellentesque vulputate lectus quis pellentesque commodo. Aliquam erat volutpat. Vestibulum in egestas velit. Pellentesque fermentum nisl vitae fringilla venenatis. Etiam id mauris vitae orci maximus ultricies. Cras fringilla ipsum magna, in fringilla dui commodo a.

Etiam vehicula luctus fermentum. In vel metus congue, pulvinar lectus vel, fermentum dui. Maecenas ante orci, egestas ut aliquet sit amet, sagittis a magna. Aliquam ante quam, pellentesque ut dignissim

quis, laoreet eget est. Aliquam erat volutpat. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Ut ullamcorper justo sapien, in cursus libero viverra eget. Vivamus auctor imperdiet urna, at pulvinar leo posuere laoreet. Suspendisse neque nisl, fringilla at iaculis scelerisque, ornare vel dolor. Ut et pulvinar nunc. Pellentesque fringilla mollis efficitur. Nullam venenatis commodo imperdiet. Morbi velit neque, semper quis lorem quis, efficitur dignissim ipsum. Ut ac lorem sed turpis imperdiet eleifend sit amet id sapien.