Arrays

Count odd and even elements

Sum of the elements in the array

Largest element in the array

Largest and smallest element in the array

Reverse the array

Search an element in the array

Sum of 2d array

Sliding Windows

Introduction

Maximum sum subarray of size K

Smallest subarray with a given sum

Two Pointers

Introduction

Pair with target sum

Remove all occurrences of an element from array

Move all zeros to the end

Remove duplicates from sorted array

Remove duplicates(at most twice) from sorted array

Linked Lists

Print the linked list in reverse order

Get Nth node from the last

Detect the loop in the linked list

Reverse the linked list

Get Nth node of the linked list

Find the middle node of the linked list [using loop]

Find the middle node of the linked list [using slow & fast pointers]

Fast & Slow Pointers

Introduction

Middle of the linked list

Linked list loop

Length of linked list loop

Start of linked list loop

Happy number

Stacks

Implement two stacks in an array[method 1]

Implement two stacks in an array[method 2]

Reverse a string using Stack

Check if the given expression is balanced or not

Introduction to infix prefix postfix expressions

Evaluate the postfix expression using Stack

Cyclic Sorts

Introduction

Cyclic sort

Find the duplicate number

Find the missing number

Binary Search Trees

Find minimum value in BST

Find sizeof() BST

Find maximum depth or height of BST

Mirror of the tree

Same tree

Double tree

Check if a binary tree is BST or not

BFS

Introduction

Binary tree level order traversal

Level order successor

Minimum depth of a binary tree

DFS

Introduction

Binary tree path sum

Print paths in a tree

All paths for a sum

Linked List Reversal

Introduction

Reverse the linked list

Reverse the sublist

Reverse every K nodes of a linked list

Top K Elements

Introduction

Top K numbers using max heap

Top K numbers using min heap

Kth smallest number

K closest points to the origin

Connect N ropes with minimum cost

Dynamic Programming

Introduction to dynamic programming

Implementation of Nth fibonacci: bottom-up approach

Implementation of Nth fibonacci: to-down approach with memoization

Memoization: avoid recomputation issue in top-down approach

Characteristics of dynamic programming

Knapsack problem introduction

0-1 Knapsack

Longest common subsequence

Longest common substring

Minimum number of jumps to reach end

Convert one string to another using minimum number of operations

Count number of ways to reach n'th stair

Cutting a rod to maximise profit

Binary Search

What is binary search?

Implementation of binary search

Time and space complexity analysis

Find in mountain array

Search in rotated sorted array

Sqrt(x) using binary search

Peak index in a mountain array

Binary search using divide and conquer strategy

Search insert position of K in a sorted array

Find first and last position of element in sorted array

Find the number of occurrences of an element in a sorted array

Find minimum in rotated sorted array

Search in row wise column wise sorted matrix

Numbers

Find positive or negative number

Absolute value of a number

Odd or even problem

Swap two numbers

Sum of natural numbers

Sum of natural numbers method 2

Factor of a number

Split numbers into digits

Sum of digits

Reverse a number

Palindrome number

Armstrong number

Fibonacci series

Prime number

Perfect number

Binary Search Trees

Find minimum value in BST

Find sizeof() BST

Find maximum depth or height of BST

Mirror of the tree

Same tree

Double tree

Check if a binary tree is BST or not

Hashing

Check if an array is a subset of another array

check-for-disjoint-arrays

Check-for-subarray-with-sum-zero

Strings

String length

Count vowels in a string

Compare two strings problem

String copy

Concatenate two strings

Lower and upper problem

Reverse the string - method 1

Reverse the string - method 2

Palindrome of the string

Bitwise

Find equal or not using bitwise

Find odd or even using bitwise

Swap two numbers using bitwise

Count set bits in an integer

Clear the rightmost set bit of a number

Brian Kernighan’s algorithm (count set bits)

Enable nth bit of a number

Check nth bit is set or unset

Disable nth bit of a number

Toggle nth bit of a number

Check if a number is a power of 2

Check if the number has alternate bit pattern

Find the odd occuring numbers using bitwise

Two single numbers

Find the missing number

Bitwise XOR

Introduction

Single number

Two single number

Recursion

Introduction recursion & basic rules

How recursion works [factorial of a number]

Stack overflow in recursion

Changing iterative function to recursive function

Recursion vs Iteration

Types of recursion

Tail recursion

Head recursion

Nested recursion

Binary or tree recursion

Indirect recursion

Why tail recursion is efficient? - part1

Why tail recursion is efficient? - part2

Sum of natural numbers

Decimal to binary conversion

Modulo operation

Find string length

Reverse a string

Check if a string is a palindrome

Count vowels in a string

Sum of numbers in a singly linked list