DSA Question Bank (Beginner Friendly)

Arrays

- Find the maximum and minimum element in an array.
- Reverse an array.
- Rotate an array by K steps.
- Find the missing number in an array of 1…N.
- Find the intersection of two arrays.
- Find the union of two arrays.
- Find the duplicate element in an array.
- Move all zeroes to the end of the array.
- Find the subarray with maximum sum (Kadane’s algorithm).
- Find the subarray with a given sum.
- Left rotate an array by one place.
- Right rotate an array by one place.
- Find the frequency of each element in an array.
- Check if the array is sorted.
- Merge two sorted arrays.
- Find the majority element in an array.
- Find the second largest element in an array.
- Find the equilibrium index of an array.
- Find leaders in an array.
- Find the missing and repeating number in an array.
- Rearrange positive and negative numbers alternately.
- Sort an array of 0s, 1s, and 2s.
- Find the longest consecutive subsequence.
- Find the maximum product subarray.
- Find the common elements in three arrays.
- Find pairs with a given sum.
- Find the median of two sorted arrays.
- Find the smallest subarray with a sum greater than a given value.
- Find the maximum difference between two elements.
- Find the count of subarrays with sum equal to zero.
- Find the maximum circular subarray sum.
- Find the missing smallest positive number.
- Find triplets with a given sum.
- Find duplicates in an array in O(n) time and O(1) space.
- Find the minimum number of platforms required for trains.
- Find the inversion count in an array.
- Find the longest increasing subsequence.
- Find the maximum length of bitonic subsequence.
- Find the maximum length subarray with equal 0s and 1s.
- Find the maximum water trapped between buildings.
- Find the maximum distance between two occurrences of the same element.
- Find the count of smaller elements on the right side.
- Find the maximum length subarray with sum divisible by K.
- Find the maximum repeating number.
- Find the sum of all subarrays.
- Find the number of subarrays with product less than K.
- Find the peak element in an array.
- Find the index of the first repeating element.
- Find the first missing even number in an array.

Strings

- Check if a string is palindrome.
- Reverse a string without using library function.
- Find the first non-repeating character in a string.
- Count vowels and consonants in a string.
- Check if two strings are anagrams.
- Remove duplicate characters from a string.
- Find the frequency of each character in a string.
- Convert a string to uppercase without library function.
- Convert a string to lowercase without library function.
- Check if a string contains only digits.
- Find the longest word in a sentence.
- Count the number of words in a sentence.
- Replace spaces in a string with ‘-’.
- Check if a string is a pangram.
- Find the most frequent character in a string.
- Print all substrings of a string.
- Find the longest palindrome substring.
- Check if two strings are rotations of each other.
- Remove all vowels from a string.
- Remove all spaces from a string.
- Find the ASCII value of each character in a string.
- Convert a Roman numeral to an integer.
- Convert an integer to Roman numeral.
- Implement strstr (substring search).
- Find the first occurrence of a substring.
- Count the occurrences of a substring.
- Find the longest common prefix of strings.
- Check if a string contains another string.
- Print the reverse of each word in a string.
- Capitalize the first letter of each word.
- Find the minimum distance between two words in a string.
- Find duplicate words in a sentence.
- Remove duplicate words from a string.
- Check if two strings are isomorphic.
- Check if one string is subsequence of another.
- Find the edit distance between two strings.
- Implement string compression.
- Check if a string has balanced brackets.
- Check if a string is valid shuffle of two strings.
- Find the longest repeating subsequence in a string.
- Find the smallest window containing all characters of another string.
- Remove characters from string2 that are in string1.
- Count distinct substrings of a string.
- Check if a string can be rearranged into a palindrome.
- Check if two strings are scrambled strings.
- Find all permutations of a string.
- Check if a string contains all unique characters.
- Check if two strings differ by exactly one edit.
- Find the longest common subsequence of two strings.

Stacks

- Implement a stack using arrays.
- Implement a stack using linked list.
- Check for balanced parentheses using stack.
- Evaluate a postfix expression.
- Convert an infix expression to postfix.
- Convert an infix expression to prefix.
- Sort a stack using another stack.
- Implement two stacks in a single array.
- Find the next greater element for each array element using stack.
- Find the next smaller element for each array element using stack.
- Design a stack that supports getMin() in O(1).
- Reverse a string using stack.
- Check if a string is palindrome using stack.
- Implement a stack with push(), pop(), and middle element retrieval.
- Implement a stack with getMax() in O(1).
- Check redundant brackets in an expression.
- Implement a stack using queue.
- Sort characters of a string using stack.
- Delete the middle element of a stack.
- Reverse a stack using recursion.
- Find the largest rectangle in a histogram.
- Evaluate a prefix expression.
- Check if a stack is sorted.
- Find celebrity problem using stack.
- Implement a stack using only one queue.
- Implement push, pop, getMin using only O(1) space.
- Implement undo feature using stack.
- Reverse words in a string using stack.
- Design a browser history system using two stacks.
- Check valid parenthesis string with wildcards using stack.
- Implement Tower of Hanoi using stack.
- Remove adjacent duplicates in string using stack.
- Find maximum area of rectangle in binary matrix using stack.
- Find minimum add to make parentheses valid using stack.
- Implement stack with dynamic resizing.
- Check for duplicate parentheses.
- Evaluate boolean expression using stack.
- Check expression has balanced symbols {},[],().
- Reverse first k elements of stack.
- Check if one stack permutation is possible of another.
- Implement custom stack with size limit.
- Find span of stock prices using stack.
- Implement push(), pop() in O(1) with two stacks.
- Find max rectangle of 1s in binary matrix using stack.
- Check if stack is palindrome.
- Clone a stack with extra space.
- Clone a stack without using extra space.
- Implement LRU cache with stack.
- Find minimum element in stack without extra space.

Queues

- Implement a queue using arrays.
- Implement a queue using linked list.
- Implement a circular queue.
- Implement a priority queue.
- Implement a double-ended queue (deque).
- Implement a queue using two stacks.
- Implement a stack using two queues.
- Reverse a queue using stack.
- Reverse the first K elements of a queue.
- Generate binary numbers from 1 to N using queue.
- Implement breadth-first search using queue.
- Implement level order traversal of a tree using queue.
- Check if a queue is palindrome.
- Implement a queue with dynamic resizing.
- Find maximum element in every sliding window using deque.
- Implement a queue using one stack.
- Check if queue follows FIFO property.
- Design hit counter using queue.
- Design circular tour problem using queue.
- Design petrol pump problem using queue.
- Implement job scheduling using priority queue.
- Implement cache using queue.
- Implement message queue.
- Design round robin scheduling using queue.
- Implement queue rotation by k elements.
- Find first non-repeating character in a stream using queue.
- Find maximum sum of k consecutive elements using queue.
- Implement deque using linked list.
- Implement deque using doubly linked list.
- Implement min-max queue.
- Implement priority queue using heap.
- Implement sliding window maximum using deque.
- Check if rotations of a queue result in sorted order.
- Implement snake and ladder problem using queue.
- Implement ticket counter simulation using queue.
- Simulate CPU scheduling using queue.
- Implement queue for interleaving elements.
- Print binary tree in zig-zag order using two queues.
- Check whether a given sequence is a queue permutation of another.
- Design task scheduler using queue.
- Implement multiple queues in a single array.
- Implement bank counter simulation using queue.
- Implement buffer management using queue.
- Implement circular tour to find start petrol pump.
- Find first negative integer in every window of size k using deque.
- Design double-ended priority queue.
- Implement queue reversal using recursion.
- Find sum of maximum and minimum in every window of size k.

Searching & Sorting

- Implement linear search.
- Implement binary search.

- Find the index of first occurrence of an element.
- Find the index of last occurrence of an element.
- Search in rotated sorted array.
- Find square root of a number using binary search.
- Find peak element using binary search.
- Find floor of an element in sorted array.
- Find ceil of an element in sorted array.
- Find first and last position of element in sorted array.
- Count occurrences of an element in sorted array.
- Find single element in sorted array (all others twice).
- Search in 2D matrix.
- Find missing element using binary search.
- Find minimum element in rotated sorted array.
- Find position to insert element in sorted array.
- Find kth smallest element using sorting.
- Find kth largest element using sorting.
- Sort an array using bubble sort.
- Sort an array using selection sort.
- Sort an array using insertion sort.
- Sort an array using merge sort.
- Sort an array using quick sort.
- Sort an array using heap sort.
- Find union of two sorted arrays.
- Find intersection of two sorted arrays.
- Find median of two sorted arrays.
- Find majority element using sorting.
- Find missing number using sorting.
- Find pair with given sum in sorted array.
- Find triplet with given sum using sorting.
- Find common elements in three sorted arrays.
- Find duplicates in array using sorting.
- Find minimum difference pair in sorted array.
- Find maximum difference pair in sorted array.
- Check if array is sorted and rotated.
- Find smallest missing positive number.
- Find two elements with sum closest to zero.
- Find element occurring more than n/2 times.
- Find maximum product of three numbers.
- Implement counting sort.
- Implement radix sort.
- Implement bucket sort.
- Find kth smallest element using quick select.
- Find element in mountain array.
- Find smallest positive number missing.
- Search in infinite sorted array.
- Find position of element using binary search in infinite array.
- Find rank of an element in sorted array.