TCS NQT Coding Cheatsheet

Arrays

- **Find the Total Number of Subarrays with Given Sum**
  - **Problem:** Given an array of integers and an integer `sum`, return the total number of subarrays whose sum equals `sum`.
  - **Test Case:**
    - Input: `nums = [1, 2, 3, 1, 1, 1], sum = 3`
    - Output: `4`
- **Rotate a Matrix by 90 Degrees in Clockwise Direction**
  - **Problem:** Given an `n x n` matrix, rotate the matrix by 90 degrees in the clockwise direction. The rotation should be done in-place.
  - **Test Case:**
    - Input: `matrix = [[1,2,3], [4,5,6], [7,8,9]]`
    - Output: `[[7,4,1], [8,5,2], [9,6,3]]`
- **Find the Subarray with the Largest Sum**
  - **Problem:** Given an integer array `nums`, find the subarray with the largest sum and return its sum.
  - **Test Case:**
    - Input: `nums = [-2,1,-3,4,-1,2,1,-5,4]`
    - Output: `6`
- **Remove Duplicates from a String Array**
  - **Problem:** Given an array of strings, remove all duplicate strings and return the updated array.
  - **Test Case:**
    - Input: `arr = ["apple", "banana", "orange", "banana", "apple"]`

    - Output: `["apple", "banana", "orange"]`

- **Find Count of Elements Greater Than All Prior Elements**
    - **Problem:** Given an integer array `Arr`, find the count of elements whose value is greater than all of its prior elements.
    - **Test Case:**
        - Input: `Arr = [7,4,8,2,9]`
        - Output: `3`
- **Sort Array Based on Risk Levels**
    - **Problem:** Given an array of risk levels (integers ranging from 0 to 2), sort the array based on these risk levels.
    - **Test Case:**
        - Input: `riskLevels = [2, 0, 1, 2, 1, 0]`
        - Output: `[0, 0, 1, 1, 2, 2]`
- **Push Empty Packets to the End of the Array**
    - **Problem:** Given an array of integer values, find all empty packets (represented by `0`) and push them to the end of the array.
    - **Test Case:**
        - Input: `arr = [2, 0, 3, 0, 5]`
        - Output: `[2, 3, 5, 0, 0]`

- **Two Sum**
    - **Problem:** Find two numbers in an array that add up to a specific target.
    - **Test Case:**
        - Input: `nums = [2, 7, 11, 15], target = 9`
        - Output: `[0, 1]`
    - **Source:** LeetCode
- **Program to Check if an Array is Sorted or Not**
    - **Problem:** Determine if a given array is sorted in non-decreasing order.

  - **Test Case:**
    - Input: `arr = [1, 2, 3, 4, 5]`
    - Output: `True`
  - **Source:** GeeksforGeeks
- **Sum of Elements in a Given Array**
  - **Problem:** Calculate the sum of all elements in an array.
  - **Test Case:**
    - Input: `arr = [1, 2, 3, 4]`
    - Output: `10`
  - **Source:** GeeksforGeeks
- **Pascal's Triangle**
  - **Problem:** Generate the first few rows of Pascal's Triangle.
  - **Test Case:**
    - Input: `numRows = 5`
    - Output: `[[1], [1,1], [1,2,1], [1,3,3,1], [1,4,6,4,1]]`
  - **Source:** LeetCode
- **Counting Frequencies of Array Elements**
  - **Problem:** Count the frequency of each element in an array.
  - **Test Case:**
    - Input: `arr = [1, 2, 2, 3, 3, 3]`
    - Output: `{1: 1, 2: 2, 3: 3}`
  - **Source:** GeeksforGeeks
- **Move Zeroes**
  - **Problem:** Move all zeros in an array to the end while maintaining the order of non-zero elements.
  - **Test Case:**
    - Input: `arr = [0, 1, 0, 3, 12]`
    - Output: `[1, 3, 12, 0, 0]`

- **Source:** LeetCode

- **Add an Element to an Array**
  - **Problem:** Add a new element to the end of an array.
  - **Test Case:**
    - Input: `arr = [1, 2, 3], element = 4`
    - Output: `[1, 2, 3, 4]`
  - **Source:** GeeksforGeeks
- **Contains Duplicate**
  - **Problem:** Check if there are any duplicates in an array.
  - **Test Case:**
    - Input: `arr = [1, 2, 3, 1]`
    - Output: `True`
  - **Source:** LeetCode
- **Find Duplicates in O(n) Time and O(1) Extra Space**
  - **Problem:** Find duplicates in an array in linear time and constant space.
  - **Test Case:**
    - Input: `arr = [4, 3, 2, 7, 8, 2, 3, 1]`
    - Output: `[2, 3]`
  - **Source:** GeeksforGeeks
- **Print Array After It Is Right Rotated K Times**
  - **Problem:** Rotate an array to the right K times and print the result.
  - **Test Case:**
    - Input: `arr = [1, 2, 3, 4, 5], K = 2`
    - Output: `[4, 5, 1, 2, 3]`
  - **Source:** GeeksforGeeks
- **Single Number**
  - **Problem:** Find the element that appears only once in an array where every other element appears twice.

- **Test Case:**
  - Input: `arr = [4, 1, 2, 1, 2]`
  - Output: `4`
- **Source:** LeetCode

- **Mean and Median of an Unsorted Array**
  - **Problem:** Calculate the mean and median of an unsorted array.
  - **Test Case:**
    - Input: `arr = [1, 2, 3, 4, 5, 6, 7]`
    - Output: `Mean: 4, Median: 4`
  - **Source:** GeeksforGeeks
- **Smallest and Second Smallest Elements in an Array**
  - **Problem:** Find the smallest and second smallest elements in an array.
  - **Test Case:**
    - Input: `arr = [12, 13, 1, 10, 34, 1]`
    - Output: `Smallest: 1, Second Smallest: 10`
  - **Source:** GeeksforGeeks
- **Third Maximum Number**
  - **Problem:** Find the third maximum number in an array.
  - **Test Case:**
    - Input: `arr = [3, 2, 1]`
    - Output: `1`
  - **Source:** LeetCode
- **Sort Elements by Frequency**
  - **Problem:** Sort the elements of an array by their frequency of occurrence.
  - **Test Case:**
    - Input: `arr = [2, 5, 2, 8, 5, 6, 8, 8]`
    - Output: `[8, 8, 8, 2, 2, 5, 5, 6]`

    - **Source:** GeeksforGeeks
- **Majority Element**
    - **Problem:** Find the majority element that appears more than half the time in an array.
    - **Test Case:**
        - Input: `arr = [3, 2, 3]`
        - Output: `3`
    - **Source:** LeetCode
- **Next Greater Element I**
    - **Problem:** Find the next greater element for each element in an array.
    - **Test Case:**
        - Input: `nums1 = [4, 1, 2], nums2 = [1, 3, 4, 2]`
        - Output: `[-1, 3, -1]`
    - **Source:** LeetCode
- **Intersection of Two Arrays**
    - **Problem:** Find the intersection of two arrays.
    - **Test Case:**
        - Input: `nums1 = [1, 2, 2, 1], nums2 = [2, 2]`
        - Output: `[2]`
    - **Source:** LeetCode
- **Find Peak Element**
    - **Problem:** Find a peak element in an array where a peak is an element greater than its neighbors.
    - **Test Case:**
        - Input: `arr = [1, 2, 3, 1]`
        - Output: `2`
    - **Source:** LeetCode
- **Find Peak Element**

- **Problem:** Find a peak element in an array where a peak is an element greater than its neighbors.
- **Test Case:**
  - Input: `arr = [1, 2, 3, 1]`
  - Output: `2`
- **Source:** LeetCode

- **Longest Continuous Increasing Subsequence**
  - **Problem:** Find the length of the longest continuous increasing subsequence in an array.
  - **Test Case:**
    - Input: `arr = [1, 3, 5, 4, 7]`
    - Output: `3`
  - **Source:** LeetCode

STRINGS

- **Print a Special Pyramid from an Input Array**
  - **Problem:** Given a height `h` and an input array `arr`, print a special pyramid where numbers less than 5 must be padded with zeroes.
  - **Test Case:**
    - Input: `height = 3, arr = [6, 28, 66, 120, 190, 276]`
    - Output:

```
00006
00028 00066
00120 00190 00276
```

- **Furnishing Company Curtain Color Counting**
  - **Problem:** Given a string of curtain colors represented by `'a'` for aqua and `'b'` for black, and an integer `L`, find the number of aqua color curtains in the box with the maximum number of aqua color curtains.

  - **Test Case:**
    - Input: `str = "aabbaa", L = 3`
    - Output: `3`

- **Remove Duplicates from a String Array**
  - **Problem:** Given an array of strings, remove all duplicate strings and return the updated array.
  - **Test Case:**
    - Input: `arr = ["apple", "banana", "orange", "banana", "apple"]`
    - Output: `["apple", "banana", "orange"]`

- **Given String is Palindrome or Not**
  - **Problem:** Check if a given string is a palindrome.
  - **Test Case:**
    - Input: `str = "madam"`
    - Output: `True`
  - **Source:** GeeksforGeeks
- **Find Common Characters**
  - **Problem:** Find the common characters in multiple strings.
  - **Test Case:**
    - Input: `words = ["bella", "label", "roller"]`
    - Output: `["e", "l", "l"]`
  - **Source:** LeetCode
- **Remove Character**
  - **Problem:** Remove characters from the first string that are present in the second string.
  - **Test Case:**
    - Input: `str1 = "abcdef", str2 = "acf"`
    - Output: `"bde"`

- **Source:** GeeksforGeeks

- **Reverse a String**
  - **Problem:** Reverse a given string.
  - **Test Case:**
    - Input: `str = "hello"`
    - Output: `"olleh"`
  - **Source:** GeeksforGeeks
- **Remove All Adjacent Duplicates in String**
  - **Problem:** Remove all adjacent duplicates in a string.
  - **Test Case:**
    - Input: `str = "abbaca"`
    - Output: `"ca"`
  - **Source:** LeetCode
- **Uncommon Words from Two Sentences**
  - **Problem:** Find the uncommon words from two sentences.
  - **Test Case:**
    - Input: `s1 = "this apple is sweet", s2 = "this apple is sour"`
    - Output: `["sweet", "sour"]`
  - **Source:** LeetCode
- **Excel Sheet Column Number**
  - **Problem:** Convert a column title to a number in an Excel sheet.
  - **Test Case:**
    - Input: `columnTitle = "AB"`
    - Output: `28`
  - **Source:** LeetCode
- **Frequency of Characters in a String**
  - **Problem:** Print characters and their frequencies in order of their occurrence.

- **Test Case:**
  - Input: `str = "geeksforgeeks"`
  - Output: `g2 e4 k2 s2 f1 o1 r1`
- **Source:** GeeksforGeeks

- **Sort String of Characters**
  - **Problem:** Sort the characters in a string.
  - **Test Case:**
    - Input: `str = "cba"`
    - Output: `"abc"`
  - **Source:** GeeksforGeeks
- **Valid Anagram**
  - **Problem:** Check if two strings are anagrams.
  - **Test Case:**
    - Input: `str1 = "anagram", str2 = "nagaram"`
    - Output: `True`
  - **Source:** LeetCode
- **Convert Characters of a String to Opposite Case**
  - **Problem:** Convert each character in the string to its opposite case.
  - **Test Case:**
    - Input: `str = "Hello"`
    - Output: `"hELLO"`
  - **Source:** GeeksforGeeks
- **Count Binary Substrings**
  - **Problem:** Count the number of non-empty substrings that have the same number of 0's and 1's.
  - **Test Case:**
    - Input: `s = "00110011"`
    - Output: `6`

- **Source:** LeetCode

- **Common Subsequence in Two Strings**
  - **Problem:** Count the common subsequences between two strings.
  - **Test Case:**
    - Input: `str1 = "abc", str2 = "ac"`
    - Output: `3`
  - **Source:** GeeksforGeeks
- **Count Unique Characters of All Substrings of a Given String**
  - **Problem:** Count the number of unique characters in all possible substrings of a given string.
  - **Test Case:**
    - Input: `s = "ABC"`
    - Output: `10`
  - **Source:** LeetCode
- **Find the Difference**
  - **Problem:** Find the difference between two strings where one string is the original string with an extra character.
  - **Test Case:**
    - Input: `s = "abcd", t = "abcde"`
    - Output: `"e"`
  - **Source:** LeetCode
- **Duplicates in the Input String**
  - **Problem:** Print all the duplicate characters in the input string.
  - **Test Case:**
    - Input: `str = "teststring"`
    - Output: `t2 s2`
  - **Source:** GeeksforGeeks
- **Count Vowels, Consonants, Digits, and Special Characters in a String**

- **Problem:** Count vowels, consonants, digits, and special characters in a string.
- **Test Case:**
  - Input: `str = "abc123!@#"`
  - Output: `Vowels: 1, Consonants: 2, Digits: 3, Special Characters: 3`
- **Source:** GeeksforGeeks

- **Detect Capital**
  - **Problem:** Determine if the usage of capital letters in a string is correct.
  - **Test Case:**
    - Input: `word = "USA"`
    - Output: `True`
  - **Source:** LeetCode
- **Fizz Buzz**
  - **Problem:** Print the numbers from 1 to n. But for multiples of three print “Fizz” instead of the number and for the multiples of five print “Buzz”.
  - **Test Case:**
    - Input: `n = 15`
    - Output: `["1", "2", "Fizz", "4", "Buzz", "Fizz", "7", "8", "Fizz", "Buzz", "11", "Fizz", "13", "14", "FizzBuzz"]`
  - **Source:** LeetCode
- **Check if a String is Substring of Another**
  - **Problem:** Check if one string is a substring of another.
  - **Test Case:**
    - Input: `str1 = "hello", str2 = "ll"`
    - Output: `True`
  - **Source:** GeeksforGeeks

SORTING

Bubble Sort
Selection Sort
Insertion Sort
Merge Sort
Quick Sort

Mathematical

- **Sum of the Cubes of All Integers Within a Given Range [n, m]**
  - **Problem:** Given two integers `n` and `m`, calculate the sum of the cubes of all integers within the range `[n, m]`.
  - **Test Case:**
    - Input: `n = 1, m = 3`
    - Output: `36` (1^3 + 2^3 + 3^3)
- **Find the k-th Largest Factor of a Given Number**
  - **Problem:** Given two integers `n` and `k`, find the `k-th` largest factor of `n`. If `k` exceeds the number of factors, return `1`.
  - **Test Case:**
    - Input: `n = 12, k = 2`
    - Output: `4`
- **Toggle Bits After the Most Significant Bit**
  - **Problem:** Given a positive integer, convert it to binary, toggle all bits after and including the most significant bit, and return the resultant integer.
  - **Test Case:**
    - Input: `n = 10`
    - Output: `5`
- **Armstrong Number Check**

- **Problem:** Given an integer `n`, return `True` if it is an Armstrong number, otherwise return `False`.
- **Test Case:**
  - Input: `n = 153`
  - Output: `True`

- **Calculate the Price of a Product from Its Digits**
  - **Problem:** Given a product code represented by an integer `N`, calculate the price of the product by multiplying all its digits.
  - **Test Case:**
    - Input: `N = 234`
    - Output: `24` (2 * 3 * 4)
- **Single Digit Sum After Repeated Summation**
  - **Problem:** Given an integer `N` and a repetition count `R`, calculate the single-digit sum after summing the digits of `N` `R` times.
  - **Test Case:**
    - Input: `N = 9875, R = 2`
    - Output: `2`
- **Minimum Page Turns in a Book**
  - **Problem:** Given a book with `n` pages, and a page number `p`, find the minimum number of pages to turn to reach page `p`.
  - **Test Case:**
    - Input: `n = 6, p = 2`
    - Output: `1`
- **Climbing Stairs**
  - **Problem:** Given `n` steps, find the number of distinct ways to reach the top.
  - **Test Case:**
    - Input: `n = 3`
    - Output: `3`

- **Source:** LeetCode

- **Check if a Given Year is a Leap Year**
  - **Problem:** Determine whether a given year is a leap year or not.
  - **Test Case:**
    - Input: `year = 2000`
    - Output: `True`
  - **Source:** GeeksforGeeks
- **Prime Numbers**
  - **Problem:** Given a number `n`, return all prime numbers up to `n`.
  - **Test Case:**
    - Input: `n = 10`
    - Output: `[2, 3, 5, 7]`
  - **Source:** Prime Numbers
- **Check if a Number is Positive, Negative, Odd, Even, or Zero**
  - **Problem:** Given an integer, determine if it's positive, negative, odd, even, or zero.
  - **Test Case:**
    - Input: `num = -3`
    - Output: `Negative, Odd`
  - **Source:** GeeksforGeeks
- **All Divisors of a Natural Number**
  - **Problem:** Given a natural number `n`, find all divisors of `n`.
  - **Test Case:**
    - Input: `n = 12`
    - Output: `[1, 2, 3, 4, 6, 12]`
  - **Source:** GeeksforGeeks
- **Convert a Number to Hexadecimal**
  - **Problem:** Convert a given integer to its hexadecimal representation.

  - **Test Case:**
    - Input: `num = 26`
    - Output: `"1a"`
  - **Source:** LeetCode
- **Valid Perfect Square**
  - **Problem:** Given a number `num`, return `True` if it's a perfect square.
  - **Test Case:**
    - Input: `num = 16`
    - Output: `True`
  - **Source:** LeetCode
- **Program to Add Two Fractions**
  - **Problem:** Given two fractions, find their sum and return it in the simplest form.
  - **Test Case:**
    - Input: `frac1 = 1/2, frac2 = 1/3`
    - Output: `5/6`
  - **Source:** GeeksforGeeks
- **Fibonacci Numbers**
  - **Problem:** Find the `n-th` Fibonacci number.
  - **Test Case:**
    - Input: `n = 5`
    - Output: `5`
  - **Source:** GeeksforGeeks
- **Add Digits**
  - **Problem:** Given an integer `num`, repeatedly add its digits until the result is a single digit.
  - **Test Case:**
    - Input: `num = 38`

    - Output: `2`
  - **Source:** LeetCode
- **Replace All '0' with '5' in an Input Integer**
  - **Problem:** Replace every occurrence of `0` with `5` in a given integer.
  - **Test Case:**
    - Input: `num = 1020`
    - Output: `1525`
  - **Source:** GeeksforGeeks
- **Perfect Number**
  - **Problem:** Determine whether a given number is a perfect number.
  - **Test Case:**
    - Input: `num = 28`
    - Output: `True`
  - **Source:** GeeksforGeeks
- **Armstrong Numbers**
  - **Problem:** Check if a number is an Armstrong number.
  - **Test Case:**
    - Input: `num = 153`
    - Output: `True`
  - **Source:** GeeksforGeeks
- **Sum of First n Natural Numbers**
  - **Problem:** Calculate the sum of the first `n` natural numbers.
  - **Test Case:**
    - Input: `n = 5`
    - Output: `15`
  - **Source:** GeeksforGeeks
- **Permutations to Arrange N Persons Around a Circular Table**

- **Problem:** Find the number of possible ways to arrange `N` persons around a circular table.
- **Test Case:**
  - Input: `N = 4`
  - Output: `6`
- **Source:** GeeksforGeeks

- **Roots of Quadratic Equation**
  - **Problem:** Given a quadratic equation in the form `ax^2 + bx + c = 0`, find its roots.
  - **Test Case:**
    - Input: `a = 1, b = -3, c = 2`
    - Output: `Roots = 2, 1`
  - **Source:** GeeksforGeeks
- **Maximum Product of Three Numbers**
  - **Problem:** Given an integer array, find the maximum product of three numbers in the array.
  - **Test Case:**
    - Input: `nums = [-10, -10, 5, 2]`
    - Output: `500`
  - **Source:** LeetCode
- **Happy Number**
  - **Problem:** Given a number `n`, determine if it's a happy number.
  - **Test Case:**
    - Input: `n = 19`
    - Output: `True`
  - **Source:** LeetCode

ADVANCED

- **Round Table Seating Arrangement with Constraints**
  - **Problem:** Given `N` members sitting around a circular table, find the possible number of ways to arrange them such that the president and prime minister of India are always seated next to each other.
  - **Test Case:**
    - Input: `N = 5`
    - Output: `48`
- **Sorting Confiscated Items by Risk Level**
  - **Problem:** Given an array of risk levels (integers ranging from 0 to 2), sort the array based on these risk levels.
  - **Test Case:**
    - Input: `riskLevels = [2, 0, 1, 2, 1, 0]`
    - Output: `[0, 0, 1, 1, 2, 2]`
- **International Conference Round Table Seating**
  - **Problem:** Find the possible number of ways to arrange `N` members around a circular table such that the president and prime minister of India are always seated next to each other.
  - **Test Case:**
    - Input: `N = 6`
    - Output: `240`
- **Calculate Shipping Cost**
  - **Problem:** Calculate the total shipping cost based on the package's weight and the distance it needs to travel.
  - **Test Case:**
    - Input: `weight = 10kg, distance = 100km`
    - Output: `$15`
- **Counting Sundays within a Given Number of Days**
  - **Problem:** Given a number of days `n`, count how many Sundays occur within those days, considering any start day of the month.
  - **Test Case:**

    - Input: `n = 30`
    - Output: `4`
- **Detect Loop in a Linked List**
  - **Problem:** Check whether a linked list contains a loop.
  - **Test Case:**
    - Input: Linked List: `1 -> 2 -> 3 -> 4 -> 2 (loop back)`
    - Output: `True`
  - **Source:** Detect Loop in a Linked List
- **Print the Middle of a Given Linked List**
  - **Problem:** Print the middle element of a linked list.
  - **Test Case:**
    - Input: Linked List: `1 -> 2 -> 3 -> 4 -> 5`
    - Output: `3`
  - **Source:** GeeksforGeeks
- **Can We Reverse a Linked List in Less than O(n)?**
  - **Problem:** Discuss if it's possible to reverse a linked list in less than O(n) time.
  - **Test Case:**
    - Input: Linked List: `1 -> 2 -> 3 -> 4 -> 5`
    - Output: `No`
  - **Source:** GeeksforGeeks
- **Linear Search**
  - **Problem:** Search for a target element in a given array using the linear search method.
  - **Test Case:**
    - Input: `arr = [2, 4, 0, 1, 9], target = 1`
    - Output: `3`
  - **Source:** GeeksforGeeks

- **Binary Search**
  - **Problem:** Search for a target element in a sorted array using the binary search method.
  - **Test Case:**
    - Input: `arr = [1, 2, 3, 4, 5, 6, 7], target = 4`
    - Output: `3`