LinkedIn

1. Given a 2D board and a word, find if the word exists in the grid.
   The word can be constructed from letters of sequentially adjacent cells, where "adjacent" cells are horizontally or vertically neighbouring. The same letter cell may not be used more than once.
   Example:
   Input: board = [["A","B","C","E"], ["S","F","C","S"], ["A","D","E","E"]], word = "ABCCED"
   Output: true

2. Given a string, sort it in decreasing order based on the frequency of characters.
   Example:
   Input: "tree"
   Output: "eert"
   Explanation:
   'e' appears twice while 'r' and 't' both appear once.
   So 'e' must appear before both 'r' and 't'. Therefore "eetr" is also a valid answer.

3. Given an m x n 2d grid map of '1's (land) and '0's (water), return the number of islands.
   An island is surrounded by water and is formed by connecting adjacent lands horizontally or vertically. You may assume all four edges of the grid are all surrounded by water.
   Example:
   Input: grid = [
   ["1","1","1","1","0"],
   ["1","1","0","1","0"],
   ["1","1","0","0","0"],
   ["0","0","0","0","0"]
   ]
   Output: 1

4. You are given coins of different denominations and a total amount of money amount. Write a function to compute the fewest number of coins that you need to make up that amount. If that amount of money cannot be made up by any combination of the coins, return -1.
   You may assume that you have an infinite number of each kind of coin.
   Example:
   Input: coins = [1,2,5], amount = 11
   Output: 3
   Explanation: 11 = 5 + 5 + 1

5. Find the kth largest element in an unsorted array. Note that it is the kth largest element in the sorted order, not the kth distinct element.
   Example:
   Input: [3,2,1,5,6,4] and k = 2
   Output: 5

6. Given a collection of intervals, merge all overlapping intervals.
   Example:
   Input: intervals = [[1,3], [2,6], [8,10], [15,18]]

Output: [[1,6], [8,10], [15,18]]
Explanation: Since intervals [1,3] and [2,6] overlaps, merge them into [1,6].

7. Given an array of integers nums and an integer target, return indices of the two numbers such that they add up to target.
   You may assume that each input would have exactly one solution, and you may not use the same element twice. You can return the answer in any order.
   Example:
   Input: nums = [2,7,11,15], target = 9
   Output: [0,1]

8. Suppose you have a long flowerbed in which some of the plots are planted and some are not. However, flowers cannot be planted in adjacent plots - they would compete for water and both would die.
   Given a flowerbed (represented as an array containing 0 and 1, where 0 means empty and 1 means not empty), and a number n, return if n new flowers can be planted in it without violating the no-adjacent-flowers rule.
   Example:
   Input: flowerbed = [1,0,0,0,1], n = 1
   Output: True

9. Given an array consists of non-negative integers, your task is to count the number of triplets chosen from the array that can make triangles if we take them as side lengths of a triangle.
   Example:
   Input: [2,2,3,4]
   Output: 3
   Explanation:
   Valid combinations are:
   2,3,4 (using the first 2)
   2,3,4 (using the second 2)
   2,2,3
   Note:
   - The length of the given array won't exceed 1000.
   - The integers in the given array are in the range of [0, 1000].

10. Given two strings s and t, determine if they are isomorphic.
    Two strings are isomorphic if the characters in s can be replaced to get t.
    All occurrences of a character must be replaced with another character while preserving the order of characters. No two characters may map to the same character but a character may map to itself.
    Example:
    Input: s = "egg", t = "add"
    Output: true

11. Given a string S, find the number of different non-empty palindromic subsequences in S, and return that number modulo 10^9 + 7.
    A subsequence of a string S is obtained by deleting 0 or more characters from S. A sequence is palindromic if it is equal to the sequence reversed.

Two sequences **A_1, A_2, ...** and **B_1, B_2, ...** are different if there is some **i** for which **A_i != B_i.**
Example:
Input: S = 'bccb'
Output: 6
Explanation:
The 6 different non-empty palindromic subsequences are 'b', 'c', 'bb', 'cc', 'bcb', 'bccb'. Note that 'bcb' is counted only once, even though it occurs twice.

12. Given a list of sorted characters letters containing only lowercase letters, and given a target letter target, find the smallest element in the list that is larger than the given target. Letters also wrap around. For example, if the target is target = 'z' and letters = ['a', 'b'], the answer is 'a'.
Example:
Input: letters = ["c", "f", "j"]
target = "a"
Output: "c"

13. Roman numerals are represented by seven different symbols: I, V, X, L, C, D and M.

Symbol	Value
I	1
V	5
X	10
L	50
C	100
D	500
M	1000

For example, 2 is written as II in Roman numeral, just two one's added together. 12 is written as XII, which is simply X + II. The number 27 is written as XXVII, which is XX + V + II.

Roman numerals are usually written largest to smallest from left to right. However, the numeral for four is not IIII. Instead, the number four is written as IV. Because the one is before the five we subtract it making four. The same principle applies to the number nine, which is written as IX. There are six instances where subtraction is used:

- I can be placed before V (5) and X (10) to make 4 and 9.
- X can be placed before L (50) and C (100) to make 40 and 90.
- C can be placed before D (500) and M (1000) to make 400 and 900.

Given a roman numeral, convert it to an integer.

Example:
Input: s = "III"
Output: 3

14. All DNA is composed of a series of nucleotides abbreviated as 'A', 'C', 'G', and 'T', for example: "ACGAATTCCG". When studying DNA, it is sometimes useful to identify repeated sequences within the DNA.

Write a function to find all the 10-letter-long sequences (substrings) that occur more than once in a DNA molecule.
Example:

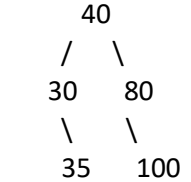
Input: s = "AAAAACCCCCAAAAACCCCCCAAAAAGGGTTT"
Output: ["AAAAACCCCC","CCCCCAAAAA"]

15. Given a string txt[0..n-1] and a pattern pat[0..m-1], write a function search(char pat[], char txt[]) that prints all occurrences of pat[] in txt[]. You may assume that n > m.
Example:
Input: txt[] = "ABABDABACDABABCABAB";
pat[] = "ABABCABAB""
Output: 10
Explanation: Pattern starts at index 10.

16. A number N is given. We need to print its 'K'th Least Significant Bit.
Example:
Input: num = 10, k = 4
Output: 1
Explanation: Binary Representation of 10 is 1010. 4th LSB is 1.

17. Given an array arr[] of size N and an integer K, the task is to find the count of distinct pairs in the array whose sum is equal to K.
Example:
Input: arr[] = { 5, 6, 5, 7, 7, 8 }
K = 13
Output: 2
Explanation:
Pairs with sum K(= 13) are { (arr[0], arr[5]), (arr[1], arr[3]), (arr[1], arr[4]) }, i.e. {(5, 8), (6, 7), (6, 7)}.
Therefore, distinct pairs with sum K(= 13) are { (arr[0], arr[5]), (arr[1], arr[3]) }.
Therefore, the required output is 2.

18. Given an array arr[] of N nodes representing preorder traversal of some BST. You have to build the exact post order from the given preorder traversal.
In Pre-Order traversal, the root node is visited before the left child and right child nodes.
Example:
Input: N = 5
arr[] = {40,30,35,80,100}
Output: 35 30 100 80 40
Explanation: PreOrder: 40 30 35 80 100
Therefore, the BST will be:



Hence, the postOrder traversal will be: 35 30 100 80 40

19. Given a String, find the longest palindromic subsequence.
    NOTE: Subsequence of a given sequence is a sequence that can be derived from the given sequence by deleting some or no elements without changing the order of the remaining elements
    Example:
    Input: S = "bbabcbcab"
    Output: 7

20. Given an integer array nums, return all the triplets [nums[i], nums[j], nums[k]] such that i != j, i != k, and j != k, and nums[i] + nums[j] + nums[k] == 0.
    Notice that the solution set must not contain duplicate triplets.
    Example:
    Input: nums = [-1,0,1,2,-1,-4]
    Output: [[-1,-1,2],[-1,0,1]]
    Explanation:
    nums[0] + nums[1] + nums[2] = (-1) + 0 + 1 = 0.
    nums[1] + nums[2] + nums[4] = 0 + 1 + (-1) = 0.
    nums[0] + nums[3] + nums[4] = (-1) + 2 + (-1) = 0.
    The distinct triplets are [-1,0,1] and [-1,-1,2].
    Notice that the order of the output and the order of the triplets does not matter.