PROBLEMS OF THE DAY – 4

1.Reorganize The Array

Given an array of elements arr[] with indices ranging from 0 to arr.size() - 1, your task is to write a program that rearranges the elements of the array such that arr[i] = i. If an element i is not present in the array, -1 should be placed at the corresponding index.

Examples:

Input: arr[] = [-1, -1, 6, 1, 9, 3, 2, -1, 4, -1]

Output: [-1, 1, 2, 3, 4, -1, 6, -1, -1, 9]

Explanation: Here We can see there are 10 elements. So, the sorted array will look like [0, 1, 2, 3, 4, 5, 6, 7, 8, 9] but in our array we are not having 0, 5, 7 and 8. So, at there places we will be printing -1 and otherplaces will be having elements.

Input: arr[] = [2, 0, 1]

Output: [0, 1, 2]

Explanation: Here We can see all the elements are present so no -1 is returned in array.

Expected Time Complexity: **O(n).**

Expected Auxiliary Space: **O(1).**

2.Second Largest

Given an array arr, return the second largest distinct element from an array. If the second largest element doesn't exist then return -1.

Examples:

Input: arr = [12, 35, 1, 10, 34, 1]

Output: 34

Explanation: The largest element of the array is 35 and the second largest element is 34.

Input: arr = [10, 10]

Output: -1

Explanation: The largest element of the array is 10 and the second largest element does not exist..

Expected Time Complexity: **O(n)**

Expected Auxiliary Space: **O(1)**

3.Trapping Rain Water

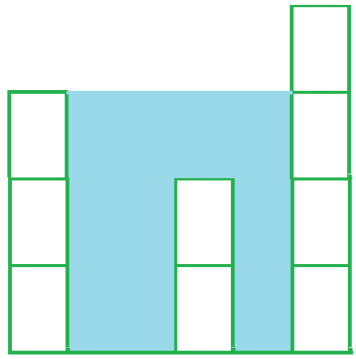
Given an array arr[] with non-negative integers representing the height of blocks. If width of each block is 1, compute how much water can be trapped between the blocks during the rainy season.

Examples:

Input: arr[] = [3,0,0,2,0,4]

Output: 10

Explanation:



Bars for input {3, 0, 0, 2, 0, 4}
Total trapped water = 3 + 3 + 1 + 3 = 10

Input: arr[] = [7,4,0,9]

Output: 10

Explanation:

Water trapped by above

block of height 4 is 3 units and above

block of height 0 is 7 units. So, the

total unit of water trapped is 10 units.

Input: arr[] = [6,9,9]

Output: 0

Explanation:

No water will be trapped.

Expected Time Complexity: **O(N)**

Expected Auxiliary Space: **O(N)**

4.nCr

Given two integers n and r, find nCr. Since the answer may be very large, calculate the answer modulo 109+7.

Note : If r is greater than n, return 0.

Example:

Input: n = 3, r = 2

Output: 3

Explaination: 3C2 = 3.

Input: n = 2, r = 4

Output: 0

Explaination: r is greater than n.

Expected Time Complexity: **O(n*r)**

Expected Auxiliary Space: **O(r)**

5.Permutations of a given string

Given a string S. The task is to print all unique permutations of the given string that may contain dulplicates in lexicographically sorted order.

Example:

Input: ABC

Output:

ABC ACB BAC BCA CAB CBA

Input: ABSG

Output:

ABGS ABSG AGBS AGSB ASBG ASGB BAGS

BASG BGAS BGSA BSAG BSGA GABS GASB

GBAS GBSA GSAB GSBA SABG SAGB SBAG

SBGA SGAB SGBA

Expected Time Complexity: **O(n! * n)**

Expected Space Complexity: **O(n! * n)**