Name:T.Harshitha

Date:10/01/2023

Python Codes

Problem Statement 1

Median of Two Sorted Arrays

Given two sorted arrays nums1 and nums2 of size m and n respectively, return **the median** of the two sorted arrays.

The overall run time complexity should be O(log (m+n)).

**Example 1:**

**Input:** nums1 = [1,3], nums2 = [2]

**Output:** 2.00000

**Explanation:** merged array = [1,2,3] and median is 2.

**Example 2:**

**Input:** nums1 = [1,2], nums2 = [3,4]

**Output:** 2.50000

**Explanation:** merged array = [1,2,3,4] and median is (2 + 3) / 2 = 2.5.

Source Code:

def findMedianSortedArrays(nums1,nums2) :

        i=0

        j=0

        output=[]

        while i<len(nums1) and j<len(nums2):

            if nums1[i]<nums2[j]:

                output.append(nums1[i])

                i=i+1

            else:

                output.append(nums2[j])

                j=j+1

        while i<len(nums1):

            output.append(nums1[i])

            i=i+1

        while j<len(nums2):

            output.append(nums2[j])

            j=j+1

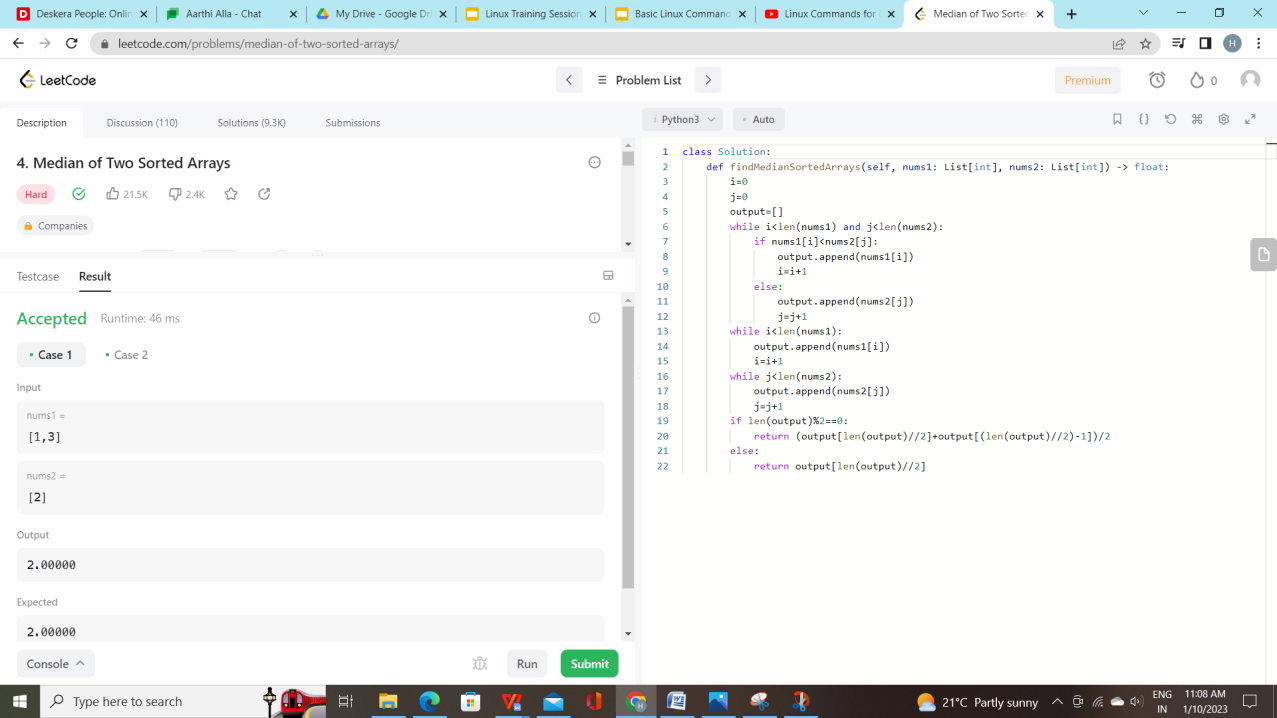
        if len(output)%2==0:

            return (output[len(output)//2]+output[(len(output)//2)-1])/2

        else:

            return output[len(output)//2]

Output:



Problem Statement 2

Merge Sorted Array

You are given two integer arrays nums1 and nums2, sorted in **non-decreasing order**, and two integers m and n, representing the number of elements in nums1 and nums2 respectively.

**Merge** nums1 and nums2 into a single array sorted in **non-decreasing order**.

The final sorted array should not be returned by the function, but instead be stored inside the array nums1. To accommodate this, nums1 has a length of m + n, where the first m elements denote the elements that should be merged, and the last n elements are set to 0 and should be ignored. nums2 has a length of n.

**Example 1:**

**Input:** nums1 = [1,2,3,0,0,0], m = 3, nums2 = [2,5,6], n = 3

**Output:** [1,2,2,3,5,6]

**Explanation:** The arrays we are merging are [1,2,3] and [2,5,6].

The result of the merge is [1,2,2,3,5,6] with the underlined elements coming from nums1.

**Example 2:**

**Input:** nums1 = [1], m = 1, nums2 = [], n = 0

**Output:** [1]

**Explanation:** The arrays we are merging are [1] and [].

The result of the merge is [1].

**Example 3:**

**Input:** nums1 = [0], m = 0, nums2 = [1], n = 1

**Output:** [1]

**Explanation:** The arrays we are merging are [] and [1].

The result of the merge is [1].

Note that because m = 0, there are no elements in nums1. The 0 is only there to ensure the merge result can fit in nums1.

Source code:

def merge(nums1,m, nums2, n) :

        k = len(nums1) - 1

        while m-1 >= 0 and n-1 >= 0:

            if nums1[m-1] > nums2[n-1]:

                nums1[k] = nums1[m-1]

                m -= 1

            else:

                nums1[k] = nums2[n-1]

                n -= 1

            k -= 1

        while n-1 >= 0:

            nums1[k] = nums2[n-1]

            n -= 1

Output:

