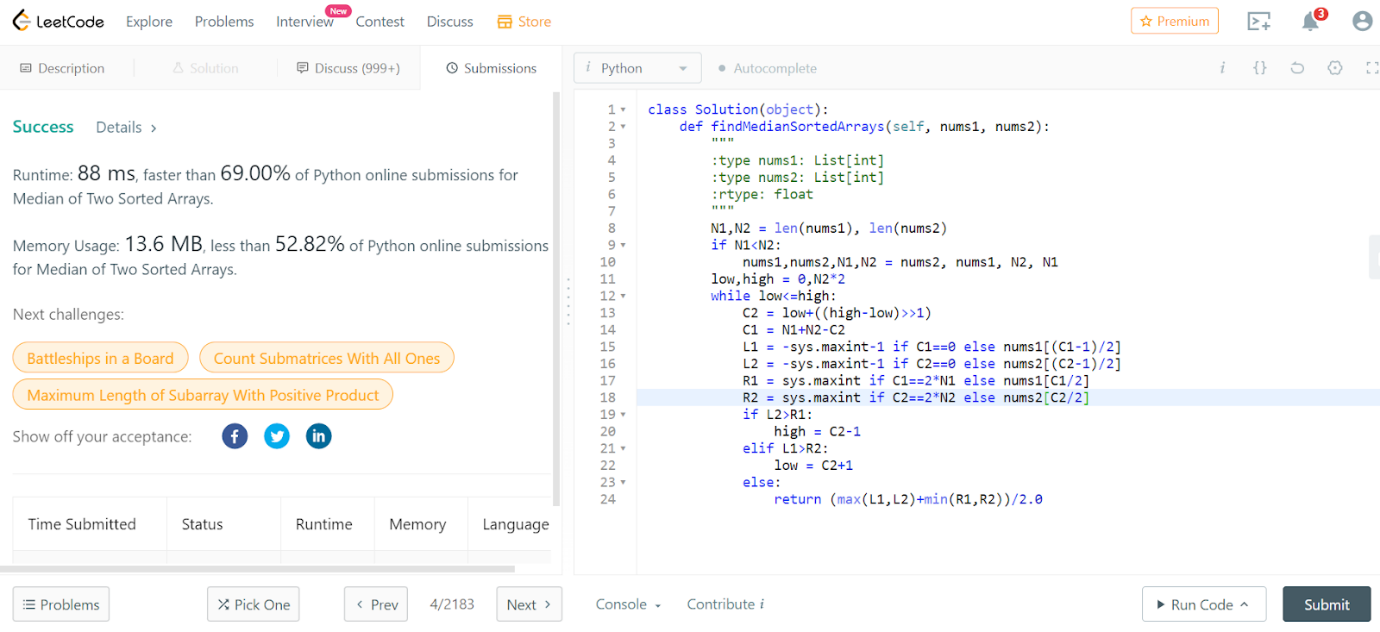
**4.**[**Median of Two Sorted Arrays**](https://leetcode.com/problems/median-of-two-sorted-arrays)



class Solution(object):

    def findMedianSortedArrays(self, nums1, nums2):

        """

        :type nums1: List[int]

        :type nums2: List[int]

        :rtype: float

        """

        N1,N2 = len(nums1), len(nums2)

        if N1<N2:

            nums1,nums2,N1,N2 = nums2, nums1, N2, N1

        low,high = 0,N2\*2

        while low<=high:

            C2 = low+((high-low)>>1)

            C1 = N1+N2-C2

            L1 = -sys.maxint-1 if C1==0 else nums1[(C1-1)/2]

            L2 = -sys.maxint-1 if C2==0 else nums2[(C2-1)/2]

            R1 = sys.maxint if C1==2\*N1 else nums1[C1/2]

            R2 = sys.maxint if C2==2\*N2 else nums2[C2/2]

            if L2>R1:

                high = C2-1

            elif L1>R2:

                low = C2+1

            else:

                return (max(L1,L2)+min(R1,R2))/2.0

**41. First Missing Positive**

Graphical user interface, text, application, email

Description automatically generated

class Solution(object):

    def firstMissingPositive(self, nums):

        for i in range(len(nums)):

            while 0 <= nums[i] - 1 < len(nums) and nums[nums[i] - 1] != nums[i]:

                tmp = nums[i] - 1

                nums[i], nums[tmp] = nums[tmp], nums[i]

        for i in range(len(nums)):

            if nums[i] != i + 1:

                return i + 1

        return len(nums) + 1