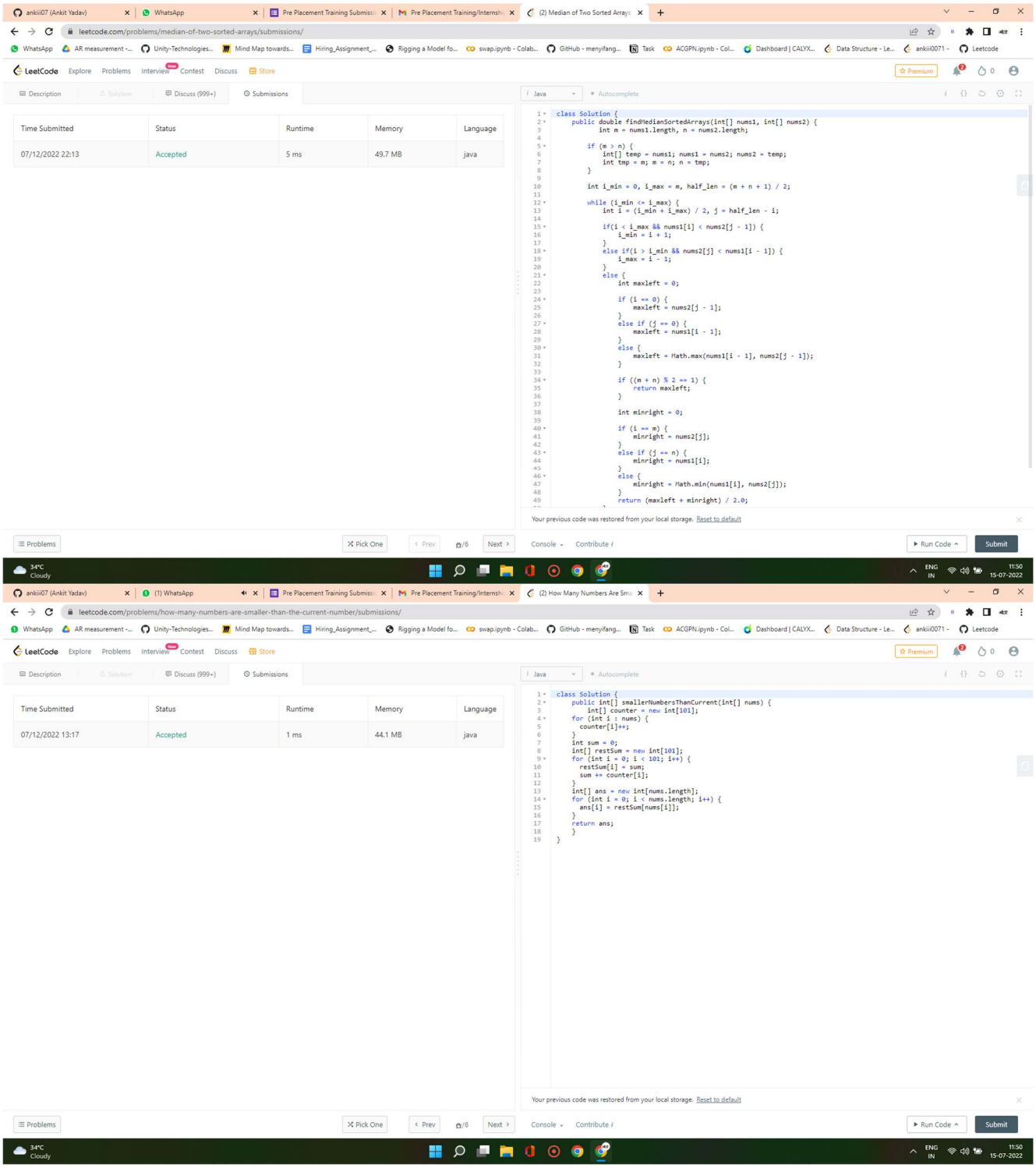
LeetCode Solution



leetcode.com/problems/median-of-two-sorted-arrays/submissions/

Time Submitted	Status	Runtime	Memory	Language
07/12/2022 22:13	Accepted	5 ms	49.7 MB	java

```java
class Solution {
    public double findMedianSortedArrays(int[] nums1, int[] nums2) {
        int m = nums1.length, n = nums2.length;

        if (m > n) {
            int[] temp = nums1; nums1 = nums2; nums2 = temp;
            int tmp = m; m = n; n = tmp;
        }

        int i_min = 0, i_max = m, half_len = (m + n + 1) / 2;

        while (i_min <= i_max) {
            int i = (i_min + i_max) / 2, j = half_len - i;

            if(i < i_max && nums1[i] < nums2[j - 1]) {
                i_min = i + 1;
            }
            else if(i > i_min && nums2[j] < nums1[i - 1]) {
                i_max = i - 1;
            }
            else {
                int maxleft = 0;

                if (i == 0) {
                    maxleft = nums2[j - 1];
                }
                else if (j == 0) {
                    maxleft = nums1[i - 1];
                }
                else {
                    maxleft = Math.max(nums1[i - 1], nums2[j - 1]);
                }

                if ((m + n) % 2 == 1) {
                    return maxleft;
                }

                int minright = 0;

                if (i == m) {
                    minright = nums2[j];
                }
                else if (j == n) {
                    minright = nums1[i];
                }
                else {
                    minright = Math.min(nums1[i], nums2[j]);
                }
                return (maxleft + minright) / 2.0;
```

Your previous code was restored from your local storage. Reset to default

Problems | Pick One | Prev | 6 | Next | Console | Contribute | Run Code | Submit

leetcode.com/problems/how-many-numbers-are-smaller-than-the-current-number/submissions/

Time Submitted	Status	Runtime	Memory	Language
07/12/2022 13:17	Accepted	1 ms	44.1 MB	java

```java
class Solution {
    public int[] smallerNumbersThanCurrent(int[] nums) {
        int[] counter = new int[101];
    for (int i : nums) {
      counter[i]++;
    }
    int sum = 0;
    int[] restSum = new int[101];
    for (int i = 0; i < 101; i++) {
      restSum[i] = sum;
      sum += counter[i];
    }
    int[] ans = new int[nums.length];
    for (int i = 0; i < nums.length; i++) {
      ans[i] = restSum[nums[i]];
    }
    return ans;
    }
}
```

Your previous code was restored from your local storage. Reset to default

Problems | Pick One | Prev | 6 | Next | Console | Contribute | Run Code | Submit