Problem 02:

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.

**Constraints:**

* nums1.length == m
* nums2.length == n
* 0 <= m <= 1000
* 0 <= n <= 1000
* 1 <= m + n <= 2000
* -106 <= nums1[i], nums2[i] <= 106

**Code:**

class Solution {

public double findMedianSortedArrays(int[] nums1, int[] nums2) {

int l1 = nums1.length;

int l2 = nums2.length;

int l3 = l1 + l2;

int[] nums3 = new int[l3];

int resodd=0, reseven1=0, reseven2=0, reseven=0, temp=0, temp1=0, temp2=0;

double a, b =0;

for(int i=0 ; i<l1; i++){

for( int j = i+1; j < l1; j++) {

if(nums1[i] > nums1[j]) {

temp = nums1[i];

nums1[i] = nums1[j];

nums1[j] = temp;

}

}

}

for(int i=0 ; i<l2; i++){

for( int j = i+1; j < l2; j++) {

if(nums2[i] > nums2[j]) {

temp1 = nums2[i];

nums2[i] = nums2[j];

nums2[j] = temp1;

}

}

}

for(int i=0; i<l1 ; i++){

nums3[i] = nums1[i];

}

for(int i=0; i<l2;i++){

nums3[l1+i] = nums2[i];

}

for(int i=0 ; i<l3; i++){

for( int j = i+1; j < l3; j++) {

if(nums3[i] > nums3[j]) {

temp2 = nums3[i];

nums3[i] = nums3[j];

nums3[j] = temp2;

}

}

}

if(l3%2 == 0){

reseven1 = l3/2;

reseven2 = reseven1-1;

a = nums3[reseven1] + nums3[reseven2];

b = a/2;

return b;

}

else{

resodd = l3/2;

return nums3[resodd];

}

}

}