

Median of Two Sorted Arrays

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

The overall runtime complexity should be $O(\log(m+n))$

Ex:

Input: $\text{nums1} = [1, 3]$, $\text{nums2} = [2]$

Output: 2.00000

constraints:

- $\text{nums1.length} == m$
- $\text{nums2.length} == n$
- $0 \leq m \leq 1000$
- $0 \leq n \leq 1000$
- $1 \leq m+n \leq 2000$
- $-10^6 \leq \text{nums1}[i], \text{nums2}[i] \leq 10^6$

Algorithm:

→ Input two sorted integer arrays num1 & num2

→ $n = \text{num1.length} + \text{num2.length}$

→ create new array of size n to store all elements of both arrays

→ copy all elements of num1 and num2 into array

- use Arrays.sort(arr) to sort the combined array
- if n is odd, return middle element $\rightarrow arr[n/2]$
- If n is even, return the average of two middle elements

$$(arr[n/2 - 1] + arr[n/2]) / 2.0$$

- Return the median value as a double.

code:

```
import java.util.Arrays;
import java.util.Scanner;
```

class Solution {

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

```
        int n = nums1.length + nums2.length;
```

```
        int[] arr = new int[n];
```

```
        for (int i = 0; i < nums1.length; i++) {
```

```
            arr[i] = nums1[i];
```

```
        }
```

```
        for (int i = 0; i < nums2.length; i++) {
```

```
            arr[i] = nums2[i];
```

```
        }
```

```
        Arrays.sort(arr);
```

```
        if (n % 2 != 0) {
```

```
            return arr[n/2];
```

```
        } else { return (arr[n/2 - 1] + arr[n/2]) / 2.0; }
```