

4. Median of Two Sorted Arrays

[Add to List ▾](#)[Question](#)[Editorial Solution](#)[My Submissions \(/problems/median-of-two-sorted-arrays/submissions/\)](/problems/median-of-two-sorted-arrays/submissions/)

Total Accepted: **141041** Total Submissions: **675022** Difficulty: **Hard** Contributors: **Admin**

There are two sorted arrays **nums1** and **nums2** of size m and n respectively.

Find the median of the two sorted arrays. The overall run time complexity should be $O(\log(m+n))$.

Example 1:

```
nums1 = [1, 3]
nums2 = [2]

The median is 2.0
```

Example 2:

```
nums1 = [1, 2]
nums2 = [3, 4]

The median is (2 + 3)/2 = 2.5
```

[Hide Company Tags](#)[Google \(/company/google/\)](/company/google/)[Zenefits \(/company/zenefits/\)](/company/zenefits/)[Microsoft \(/company/microsoft/\)](/company/microsoft/)[Apple \(/company/apple/\)](/company/apple/)[Yahoo \(/company/yahoo/\)](/company/yahoo/)[Dropbox \(/company/dropbox/\)](/company/dropbox/)[Adobe \(/company/adobe/\)](/company/adobe/)[Hide Tags](#)[Binary Search \(/tag/binary-search/\)](/tag/binary-search/)[Array \(/tag/array/\)](/tag/array/)[Divide and Conquer \(/tag/divide-and-conquer/\)](/tag/divide-and-conquer/)

Have you met this question in a real interview?

[Discuss \(https://discuss.leetcode.com/category/12\)](https://discuss.leetcode.com/category/12)[Top Solutions](#)[Pick One \(/problems/random-one-question/\)](/problems/random-one-question/)[C++](#)

```
1 class Solution {
2 public:
3
4
5     double findMedianSortedArrays(vector<int>& nums1, vector<int>& nums2) {
6         int n1 = nums1.size(), n2 = nums2.size();
7         if(n1 < n2) return findMedianSortedArrays(nums2, nums1);
8         if(n2 == 0) return ((double)(nums1[(n1 - 1) / 2]) + (double)(nums1[(n1 / 2)])) / 2.0;
9
10        int left = 0, right = n2 * 2; // important right need to *2
11        while(left <= right) {
12            int C2 = (right - left) / 2 + left; // mid center C2 of array B
13            int C1 = n1 + n2 - C2; // mid center C1 of array A
14            int L1 = C1 > 0 ? nums1[(C1 - 1) / 2] : INT_MIN;
15            int R1 = C1 < n1 * 2 ? nums1[C1 / 2] : INT_MAX; // Important: need to multiply 2
16            int L2 = C2 > 0 ? nums2[(C2 - 1) / 2] : INT_MIN;
17            int R2 = C2 < n2 * 2 ? nums2[C2 / 2] : INT_MAX;
18
19            if(L1 <= R2 && L2 <= R1) {
20                return ((double)max(L1,L2) + (double)min(R1, R2)) / 2.0;
21            } else if (L1 > R2) {
22                left = C2 + 1;
23            } else {
24                right = C2 - 1;
25            }
26        }
27        return -1;
28    }
29
30
31
32    // based on solution of
33    // https://discuss.leetcode.com/topic/16797/very-concise-o-log-min-m-n-iterative-solution-with
34    // -detailed-explanation/2
35    // double findMedianSortedArrays(vector<int>& nums1, vector<int>& nums2) {
36    //     int n1 = nums1.size();
37    //     int n2 = nums2.size();
```

[✉ Send Feedback \(mailto:admin@leetcode.com?subject=Feedback\)](mailto:admin@leetcode.com?subject=Feedback)[Notes](#)

```

37
38    //    if(n1 < n2) return findMedianSortedArrays(nums2, nums1);
39    //    if(n2 == 0) return ((double)nums1[(n1-1)/2]+(double)nums1[n1/2]) / 2;
40
41    //    int lo = 0, hi = n2 * 2;
42    //    while(lo <= hi) {
43    //        int mid2 = (lo + hi) / 2;
44    //        int mid1 = n1 + n2 - mid2;
45
46    //        double L1 = mid1 == 0 ? INT_MIN : nums1[(mid1-1) / 2];
47    //        double L2 = mid2 == 0 ? INT_MIN : nums2[(mid2-1) / 2];
48    //        double R1 = mid1 == n1 * 2 ? INT_MAX : nums1[mid1 / 2];
49    //        double R2 = mid2 == n2 * 2 ? INT_MAX : nums2[mid2 / 2];
50    //        // we need to satisfy L1 <= R1 && L1 <= R2 && L2 <= R1 && L2 <= R2
51    //        if(L1 > R2) lo = mid2 + 1;
52    //        else if(L2 > R1) hi = mid2 - 1;
53    //        else return (max(L1, L2) + min(R1, R2)) / 2;
54    //    }
55    //    return -1;
56    // }
57 };

```

Custom Testcase

☐

Shortcut: Command + enter

Contribute Testcase

Run Code

Submit Solution

Submission Result: Accepted (/submissions/detail/90704218/)

More Details
➤ (/submissions/detail/90704218/)

Next challenges:

(H) Split Array Largest Sum (/problems/split-array-largest-sum)

(H) Smallest Good Base (/problems/smallest-good-base)

(E) Max Consecutive Ones (/problems/max-consecutive-ones)

Share your acceptance!

229
43