

Median of Two Sorted Arrays (/problems/median-of-two-sorted-arrays/)

Submission Detail

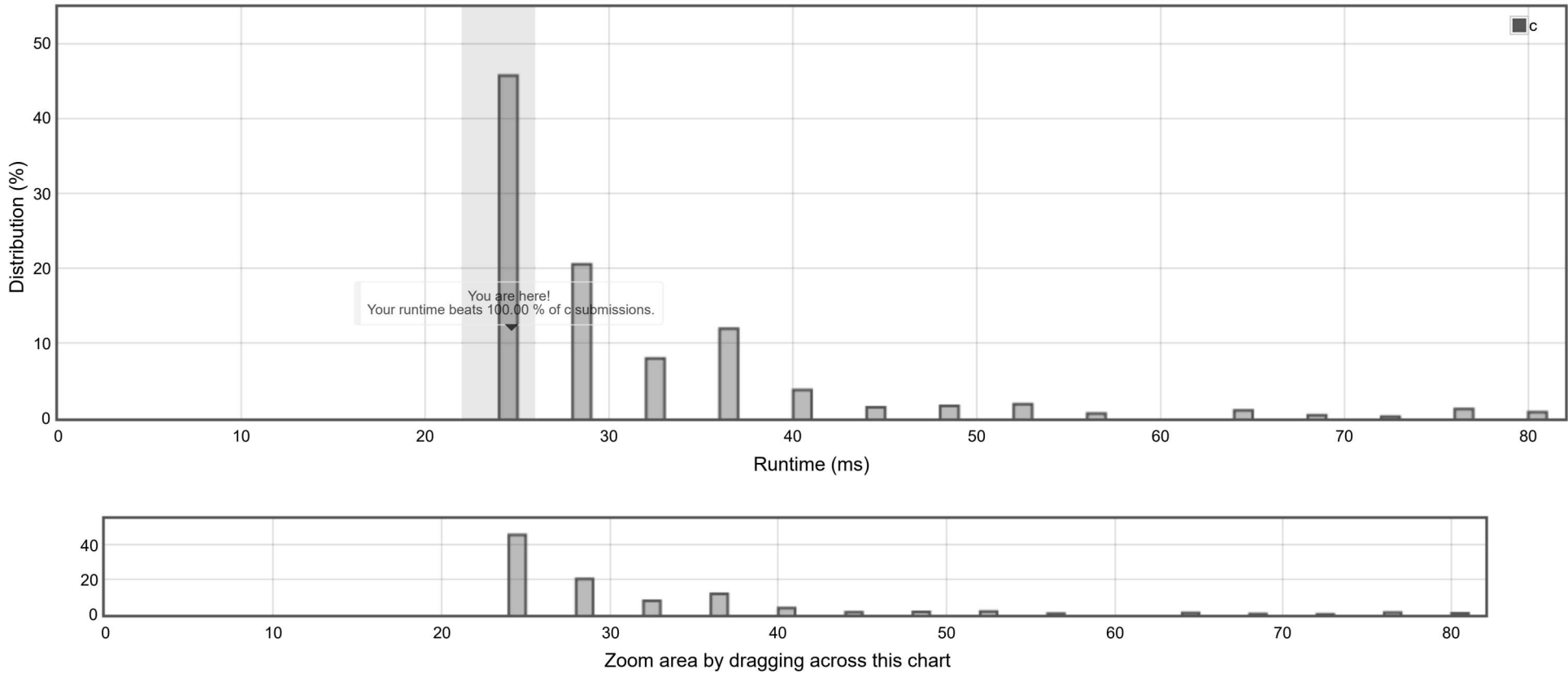
2084 / 2084 test cases passed.

Runtime: 24 ms

Status: Accepted

Submitted: 1 hour, 59 minutes ago

Accepted Solutions Runtime Distribution



Invite friends to challenge **Median of Two Sorted Arrays**

Submitted Code: 1Â hour, 59Â minutes ago

Language: c

[Edit Code](#)

```
1  #include <math.h>
2
3  int max(int a, int b) {
4
5      if (a >= b) {
6          return a;
7      }
8      return b;
9
10 }
11
12 int min(int a, int b) {
13
14     if (a <= b) {
15         return a;
16     }
17     return b;
18
19 }
20
21 void computeEdgeVals(int *nums1, int nums1Size, int *nums2, int nums2Size, int nums1LeftSize, int *maxLeft, int *minRight) {
22
23     if (nums1Size == 0) {
24         *minRight = nums2[nums2Size / 2];
25         if (nums2Size == 1) {
26             *maxLeft = *minRight;
27         } else {
28             *maxLeft = nums2[nums2Size / 2 - 1];
29         }
30         return;
31     }
32
33     if (nums2Size == 0) {
34         *minRight = nums1[nums1Size / 2];
35         if (nums1Size == 1) {
36             *maxLeft = *minRight;
37         } else {
38             *maxLeft = nums1[nums1Size / 2 - 1];
39         }
40         return;
41     }
42 }
```

```

43     int nums2LeftSize = (nums1Size + nums2Size - 2 * nums1LeftSize) / 2;
44     if (nums1LeftSize <= 0) {
45         *maxLeft = nums2[nums2LeftSize - 1];
46     } else if (nums2LeftSize <= 0) {
47         *maxLeft = nums1[nums1LeftSize - 1];
48     } else {
49         *maxLeft = max(nums1[nums1LeftSize - 1], nums2[nums2LeftSize - 1]);
50     }
51     if (nums1LeftSize == nums1Size) {
52         *minRight = nums2[nums2LeftSize];
53     } else if (nums2LeftSize == nums2Size) {
54         *minRight = nums1[nums1LeftSize];
55     } else {
56         *minRight = min(nums1[nums1LeftSize], nums2[nums2LeftSize]);
57     }
58 }
59
60
61 int judgePartition(int *nums1, int nums1Size, int *nums2, int nums2Size, int nums1LeftSize) {
62
63     int maxLeft = 0, minRight = 0;
64     int nums2LeftSize = (nums1Size + nums2Size - 2 * nums1LeftSize) / 2;
65
66     if (nums2LeftSize < 0 && nums2Size != 0) {
67         return -1;
68     }
69     if (nums2LeftSize > nums2Size && nums2Size != 0) {
70         return 1;
71     }
72
73     computeEdgeVals(nums1, nums1Size, nums2, nums2Size, nums1LeftSize, &maxLeft, &minRight);
74
75     if (maxLeft > minRight) {
76         if (maxLeft == nums1[nums1LeftSize - 1]) {
77             return -1;
78         }
79         return 1;
80     }
81     return 0;
82
83 }
84
85
86 double findMedianSortedArrays(int* nums1, int nums1Size, int* nums2, int nums2Size) {
87
88     int left = 0, right = nums1Size - 1;
89     int nums1LeftSize = nums1Size;
90     while (left <= right) {
91         int mid = (left + right) / 2;
92         int judge = judgePartition(nums1, nums1Size, nums2, nums2Size, mid + 1);
93         if (judge == -1) {


```

```
94         right = mid - 1;
95     } else if (judge == 1) {
96         left = mid + 1;
97     } else {
98         nums1LeftSize = mid + 1;
99         break;
100    }
101 }
102 if (right < 0) {
103     nums1LeftSize = 0;
104 }
105
106 int maxLeft = 0, minRight = 0;
107 computeEdgeVals(nums1, nums1Size, nums2, nums2Size, nums1LeftSize, &maxLeft, &minRight);
108 int nums2LeftSize = (nums1Size + nums2Size - 2 * nums1LeftSize) / 2;
109 if ((nums1Size + nums2Size) % 2 == 0) {
110     return ((double)maxLeft + minRight) / 2;
111 }
112 if (nums1Size == 0 || nums2Size == 0) {
113     return (double)minRight;
114 }
115 if (2 * (nums1LeftSize + nums2LeftSize) > nums1Size + nums2Size) {
116     return (double)maxLeft;
117 }
118 return (double)minRight;
119
120 }
```

[Back to problem \(/problems/median-of-two-sorted-arrays/\)](/problems/median-of-two-sorted-arrays/)

Copyright © 2018 LeetCode

[Contact Us \(/support/\)](/support/) | [Frequently Asked Questions \(/faq/\)](/faq/) | [Terms of Service \(/terms/\)](/terms/) | [Privacy Policy \(/privacy/\)](/privacy/)

 [United States \(/region/\)](/region/)