Mock Test > viniciusvramirez0108@gmail.com

Full Name:	Vinicius Valente
Email:	viniciusvramirez0108@gmail.com
Test Name:	**Mock Test**
Taken On:	7 Jul 2024 08:06:05 IST
Time Taken:	4 min 2 sec/ 10 min
Invited by:	Ankush
Invited on:	7 Jul 2024 08:05:48 IST
Skills Score:	
Tags Score:	

Algorithms	105/105
Core CS	105/105
Easy	105/105
Problem Solving	105/105
Search	105/105
Sorting	105/105
problem-solving	105/105

100%

105/105

scored in **Mock Test** in 4 min 2 sec on 7 Jul 2024 08:06:05 IST

Recruiter/Team Comments:

No Comments.

	Question Description	Time Taken	Score	Status
Q1	**Find the Median** > **Coding**	3 min 51 sec	105/ 105	✓

QUESTION 1

✓

Correct Answer

Score 105

Find the Median > Coding

Sorting Search Algorithms Easy problem-solving Core CS Problem Solving

QUESTION DESCRIPTION

The median of a list of numbers is essentially its middle element after sorting. The same number of elements occur after it as before. Given a list of numbers with an odd number of elements, find the median?

Example

$arr = [5, 3, 1, 2, 4]$

The sorted array $arr' = [1, 2, 3, 4, 5]$. The middle element and the median is 3.

Function Description

Complete the *findMedian* function in the editor below.

findMedian has the following parameter(s):

- *int arr[n]:* an unsorted array of integers

Returns

- *int:* the median of the array

Input Format

The first line contains the integer n, the size of arr.
The second line contains n space-separated integers $arr[i]$

Constraints

- $1 \leq n \leq 1000001$
- n is odd
- $-10000 \leq arr[i] \leq 10000$

Sample Input 0

```
7
0 1 2 4 6 5 3
```

Sample Output 0

```
3
```

Explanation 0

The sorted $arr = [0, 1, 2, 3, 4, 5, 6]$. It's middle element is at $arr[3] = 3$.

CANDIDATE ANSWER

Language used: **Python 3**

```
#
# Complete the 'findMedian' function below.
#
# The function is expected to return an INTEGER.
# The function accepts INTEGER_ARRAY arr as parameter.
#

def quicksort(arr):
    if len(arr) <= 1:
        return arr

    else:
        pivot = arr[len(arr)//2]
        low = [x for x in arr if x < pivot]
        same = [x for x in arr if x == pivot]
        high = [x for x in arr if x > pivot]

        return quicksort(low) + same + quicksort(high)

def findMedian(arr):
    ordered_arr = quicksort(arr)
    median = int(len(ordered_arr) / 2)

    return ordered_arr[median]
    # Write your code here
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 1	Easy	Sample case	✓ Success	0	0.0401 sec	10.3 KB
Testcase 2	Easy	Hidden case	✓ Success	35	0.0382 sec	11.2 KB

Testcase 3	Easy	Hidden case	✓ Success	35	0.0489 sec	11.9 KB
Testcase 4	Easy	Hidden case	✓ Success	35	0.1603 sec	20.8 KB

No Comments