Mock Test › khadij70707@gmail.com

Full Name:	khadij70707gmailom
Email:	khadij70707@gmail.com
Test Name:	**Mock Test**
Taken On:	13 Sep 2023 23:19:36 IST
Time Taken:	2 min 8 sec/ 10 min
Invited by:	Ankush
Invited on:	13 Sep 2023 23:19:06 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 2 min 8 sec on 13 Sep 2023 23:19:36 IST

Recruiter/Team Comments:

No Comments.

Plagiarism flagged

We have marked questions with suspected plagiarism below. Please review it in detail here -

	Question Description	Time Taken	Score	Status
Q1	Find the Median › **Coding**	2 min 1 sec	105/ 105	⚠

QUESTION 1

⚠

Needs Review

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 findMedian(arr):
    sorted_numbers = sorted(arr)

    # Calculate the length of the sorted list
    n = len(sorted_numbers)

    # Check if the list has an odd or even number of elements
    if n % 2 == 1:  # Odd number of elements
        median = sorted_numbers[n // 2]
    else:  # Even number of elements
        middle1 = sorted_numbers[n // 2 - 1]
        middle2 = sorted_numbers[n // 2]
        median = (middle1 + middle2) / 2.0

    return median

```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 1	Easy	Sample case	✓ Success	0	0.0446 sec	10.5 KB
Testcase 2	Easy	Hidden case	✓ Success	35	0.1275 sec	11.3 KB
Testcase 3	Easy	Hidden case	✓ Success	35	0.052 sec	11.9 KB
Testcase 4	Easy	Hidden case	✓ Success	35	0.1083 sec	22.3 KB

No Comments

PDF generated at: 13 Sep 2023 17:53:28 UTC