| | |
|---|---|
| Full Name: | Antrita Manduva |
| Email: | amanduva@uchicago.edu |
| Test Name: | **Mock Test** |
| Taken On: | 26 Aug 2023 20:29:53 IST |
| Time Taken: | 7 min 52 sec/ 10 min |
| Linkedin: | http://linkedin.com/antrita-manduva |
| Invited by: | Ankush |
| Invited on: | 26 Aug 2023 20:29:42 IST |
| Skills Score: | |
| Tags Score: | Algorithms 105/105<br>Core CS 105/105<br>Easy 105/105<br>Problem Solving 105/105<br>Search 105/105<br>Sorting 105/105<br>problem-solving 105/105 |

**100%**

**105/105**

scored in **Mock Test** in 7 min 52 sec on 26 Aug 2023 20:29:53 IST

**Recruiter/Team Comments:**

*No Comments.*

| | Question Description | Time Taken | Score | Status |
|---|---|---|---|---|
| Q1 | Find the Median > **Coding** | 7 min 42 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**

```python
#!/bin/python3

import math
import os
import random
import re
import sys



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

def findMedian(arr):
    arr.sort()
    if n % 2 != 0:
        med = n // 2
        return(arr[med])
    med = n / 2
    lmed, rmed = med - 1, med + 1
    return (arr[lmed] + arr[rmed]) / 2

if __name__ == '__main__':
    fptr = open(os.environ['OUTPUT_PATH'], 'w')

```

```
    n = int(input().strip())

    arr = list(map(int, input().rstrip().split()))

    result = findMedian(arr)

    fptr.write(str(result) + '\n')

    fptr.close()

```

| TESTCASE | DIFFICULTY | TYPE | STATUS | SCORE | TIME TAKEN | MEMORY USED |
|---|---|---|---|---|---|---|
| Testcase 1 | Easy | Sample case | Success | 0 | 0.0415 sec | 10.7 KB |
| Testcase 2 | Easy | Hidden case | Success | 35 | 0.0694 sec | 11.4 KB |
| Testcase 3 | Easy | Hidden case | Success | 35 | 0.0747 sec | 11.9 KB |
| Testcase 4 | Easy | Hidden case | Success | 35 | 0.0724 sec | 22 KB |

No Comments