

Full Name:

-uii Name:

Email:

Test Name:

Taken On: Time Taken:

Linkedin:

Invited by:
Invited on:

Skills Score:

Tags Score:

Antrita Manduva

amanduva@uchicago.edu

Mock Test

26 Aug 2023 20:29:53 IST

7 min 52 sec/ 10 min

http://linkedin.com/antrita-manduva

Ankush

26 Aug 2023 20:29:42 IST

Algorithms 105/105

Core CS 105/105

Easy 105/105

Problem Solving 105/105 Search 105/105

Sorting 105/105

problem-solving 105/105

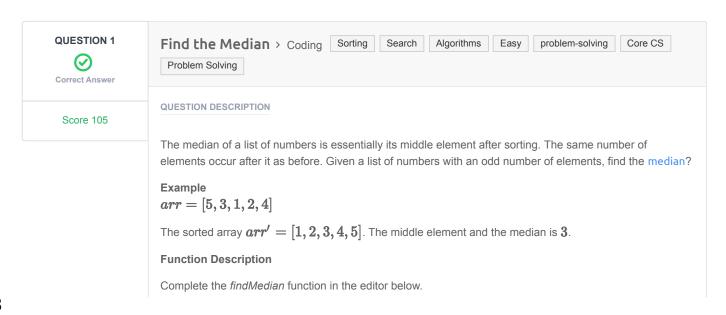
100% 105/105

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

Recruiter/Team Comments:

No Comments.





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 $m{n}$ space-separated integers $m{arr}[m{i}]$

Constraints

- $1 \le n \le 1000001$
- $oldsymbol{\cdot}$ n is odd
- $-10000 \le arr[i] \le 10000$

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

```
1 #!/bin/python3
 3 import math
 4 import os
5 import random
6 import re
7 import sys
8
11 #
12 # Complete the 'findMedian' function below.
14 # The function is expected to return an INTEGER.
15 # The function accepts INTEGER ARRAY arr as parameter.
16 #
18 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
27 if __name__ == '__main__':
       fptr = open(os.environ['OUTPUT_PATH'], 'w')
```

```
n = int(input().strip())
       arr = list(map(int, input().rstrip().split()))
       result = findMedian(arr)
34
       fptr.write(str(result) + '\n')
       fptr.close()
   TESTCASE
              DIFFICULTY
                                         STATUS
                                                    SCORE
                                                             TIME TAKEN
                                                                          MEMORY USED
                              TYPE
  Testcase 1
                                        Success
                                                              0.0415 sec
                                                                              10.7 KB
                  Easy
                           Sample case
                                                       0
  Testcase 2
                                        Success
                                                              0.0694 sec
                  Easy
                            Hidden case
                                                      35
                                                                              11.4 KB
  Testcase 3
                                                              0.0747 sec
                                                                              11.9 KB
                  Easy
                            Hidden case
                                        Success
                                                      35
  Testcase 4
                  Easy
                            Hidden case
                                        Success
                                                      35
                                                              0.0724 sec
                                                                              22 KB
No Comments
```

PDF generated at: 26 Aug 2023 15:09:40 UTC