

Mock Test > singhanand974@gmail.com

Full Name:

Anand Singh

Email:

singhanand974@gmail.com

Test Name:

Mock Test

Taken On:

25 May 2024 23:36:14 IST

Time Taken:

Contact Number:

1 min 10 sec/ 10 min

+919650875781

Linkedin:

https://www.linkedin.com/in/anand-singh-

774a2b191

Invited by:

Ankush

Invited on:

25 May 2024 23:36:06 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

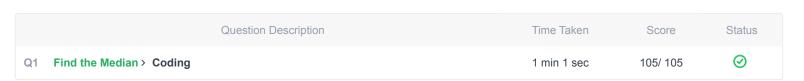
problem-solving 105/105

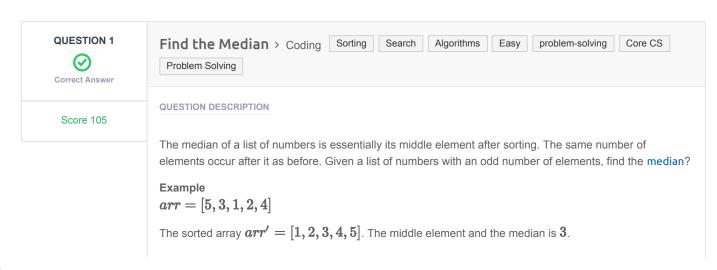


scored in **Mock Test** in 1 min 10 sec on 25 May 2024 23:36:14 IST

Recruiter/Team Comments:

No Comments.





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

- $1 \le n \le 1000001$
- *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: Java 8

```
class Result {

/*

* Complete the 'findMedian' function below.

* * The function is expected to return an INTEGER.

* The function accepts INTEGER_ARRAY arr as parameter.

*/

public static int findMedian(List<Integer> arr) {

// Write your code here

Collections.sort(arr);
int n = arr.size();
return arr.get(n/2);
}

return arr.get(n/2);
}
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 1	Easy	Sample case	Success	0	0.1003 sec	30.9 KB
Testcase 2	Easy	Hidden case	Success	35	0.15 sec	32.6 KB

	Testcase 3	Easy	Hidden case	Success	35	0.1384 sec	33.2 KB
	Testcase 4	Easy	Hidden case	Success	35	0.2583 sec	46.5 KB
N	o Comments						

PDF generated at: 25 May 2024 18:09:15 UTC