

## Mock Test > xtheanother@gmail.com

**Full Name:** Igor Kotarski Email: xtheanother@gmail.com Test Name: **Mock Test** Taken On: 27 Dec 2023 19:04:43 IST Time Taken: 8 min 6 sec/ 10 min Invited by: Ankush Invited on: 27 Dec 2023 19:02:10 IST Skills Score: Tags Score: Algorithms 105/105 Core CS 105/105

Easy 105/105

Search

Sorting

Problem Solving 105/105

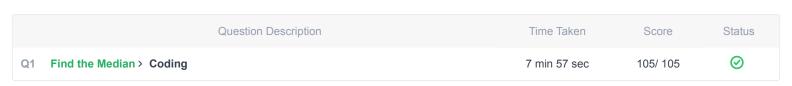
105/105 105/105

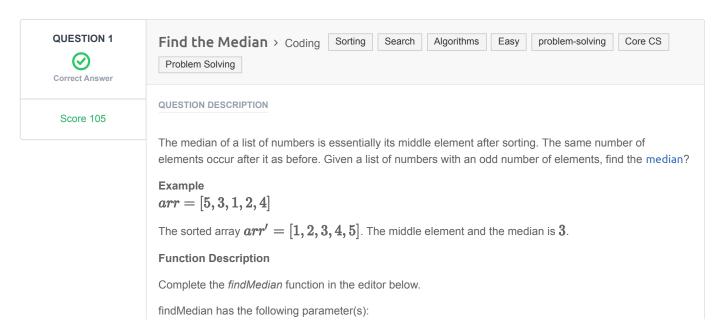
problem-solving 105/105

100% scored in Mock Test in 8 min 6 sec on 27 Dec 2023 19:04:43 IST

# **Recruiter/Team Comments:**

No Comments.





• 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: C#

```
1 class Result
2 {
4
       * 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<int> arr)
       {
          int arrLength = arr.Count();
14
          arr.Sort();
          if(arr.Count % 2 == 0)
               int number1 = arr.Count / 2;
              int number2 = (arr.Count / 2) - 1;
              return (arr[number1] + arr[number2]) / 2;
          }
          else
              return arr[arr.Count / 2];
24
       }
27 }
```

	Testcase 1	Easy	Sample case	Success	0	0.0583 sec	20.7 KB
	Testcase 2	Easy	Hidden case	Success	35	0.056 sec	21.5 KB
	Testcase 3	Easy	Hidden case	Success	35	0.0422 sec	21.6 KB
	Testcase 4	Easy	Hidden case	Success	35	0.0774 sec	29.1 KB
No Comments							

PDF generated at: 27 Dec 2023 13:44:40 UTC