

## Mock Test > gssubashofficial@gmail.com

Full Name: Subash Ganesan Email: gssubashofficial@gmail.com Test Name: **Mock Test** Taken On: 8 Apr 2024 08:27:33 IST 5 min 58 sec/ 40 min Time Taken: Invited by: Ankush 8 Apr 2024 08:27:20 IST Invited on: Skills Score: Tags Score: Algorithms 195/195 Constructive Algorithms 90/90 Core CS 195/195 Easy 105/105 Greedy Algorithms 90/90 Medium 90/90 Problem Solving 195/195 105/105 Search Sorting 105/105

100% scored in Mock Test in 5 min 58 sec on 8 Apr 2024 08:27:33 IST 195/195

# **Recruiter/Team Comments:**

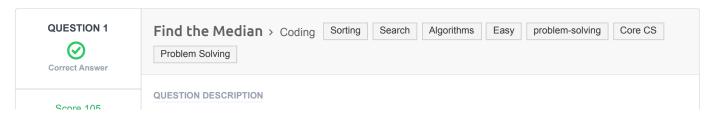
No Comments.

# **Plagiarism flagged**

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

problem-solving 195/195





00010 100

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

- $1 \le n \le 1000001$
- *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.

//

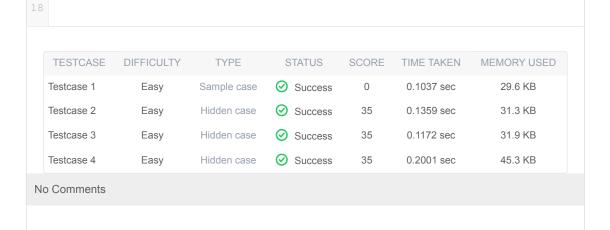
public static int findMedian(List<Integer> arr) {

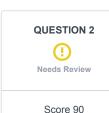
// Write your code here

Collections.sort(arr);

return arr.get(arr.size()/2);
}

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







#### QUESTION DESCRIPTION

Sean invented a game involving a  $2n \times 2n$  matrix where each cell of the matrix contains an integer. He can reverse any of its rows or columns any number of times. The goal of the game is to maximize the sum of the elements in the  $n \times n$  submatrix located in the upper-left quadrant of the matrix.

Given the initial configurations for q matrices, help Sean reverse the rows and columns of each matrix in the best possible way so that the sum of the elements in the matrix's upper-left quadrant is maximal.

# Example

$$matrix = \left[ [1,2], [3,4] \right]$$

It is  $2 \times 2$  and we want to maximize the top left quadrant, a  $1 \times 1$  matrix. Reverse row 1:

1 2

4 3

And now reverse column 0:

4 2

1 3

The maximal sum is 4.

### **Function Description**

Complete the  ${\it flippingMatrix}$  function in the editor below.

flippingMatrix has the following parameters:

- int matrix[2n][2n]: a 2-dimensional array of integers

#### Returns

- int: the maximum sum possible.

# Input Format

The first line contains an integer q, the number of queries.

The next q sets of lines are in the following format:

- The first line of each query contains an integer, n.
- Each of the next 2n lines contains 2n space-separated integers matrix[i][j] in row i of the matrix.

#### **Constraints**

- $1 \le q \le 16$
- $1 \le n \le 128$
- $0 \leq matrix[i][j] \leq 4096$ , where  $0 \leq i, j < 2n$ .

### Sample Input

# **Sample Output**

414

#### **Explanation**

Start out with the following 2n imes 2n matrix:

$$matrix = egin{bmatrix} 112 & 42 & 83 & 119 \ 56 & 125 & 56 & 49 \ 15 & 78 & 101 & 43 \ 62 & 98 & 114 & 108 \end{bmatrix}$$

Perform the following operations to maximize the sum of the  $n \times n$  submatrix in the upper-left quadrant:

2. Reverse column  $2([83, 56, 101, 114] \rightarrow [114, 101, 56, 83])$ , resulting in the matrix:

$$matrix = egin{bmatrix} 112 & 42 & 114 & 119 \ 56 & 125 & 101 & 49 \ 15 & 78 & 56 & 43 \ 62 & 98 & 83 & 108 \end{bmatrix}$$

3. Reverse row 0 ([112, 42, 114, 119]  $\rightarrow$  [119, 114, 42, 112]), resulting in the matrix:

$$matrix = egin{bmatrix} 119 & 114 & 42 & 112 \ 56 & 125 & 101 & 49 \ 15 & 78 & 56 & 43 \ 62 & 98 & 83 & 108 \end{bmatrix}$$

The sum of values in the n imes n submatrix in the upper-left quadrant is 119+114+56+125=414 .

# **CANDIDATE ANSWER**

Language used: Java 8

```
class Result {

/*

Complete the 'flippingMatrix' function below.

*
```

```
6
        * The function is expected to return an INTEGER.
        * The function accepts 2D INTEGER ARRAY matrix as parameter.
8
        */
      public static int flippingMatrix(List<List<Integer>> matrix) {
      // Write your code here
          int nx2= matrix.size();
           int nx1 = nx2/2;
          int ans = 0;
          for(int i=0; i< nx1; i++) {
               int val = 0;
               for(int j=0; j< nx1; j++){
                   val = Math.max(matrix.get(i).get(j), matrix.get(i).get(nx2-j-
19 1));
                  val = Math.max(val, matrix.get(nx2-i-1).get(j));
                   val = Math.max(val, matrix.get(nx2-i-1).get(nx2-j-1));
               ans += val;
               }
24
          }
           return ans;
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 1	Easy	Sample case	Success	0	0.0982 sec	29.6 KB
Testcase 2	Easy	Hidden case	Success	15	0.3038 sec	52 KB
Testcase 3	Easy	Hidden case	Success	15	0.3208 sec	54.3 KB
Testcase 4	Easy	Hidden case	Success	15	0.3171 sec	44.8 KB
Testcase 5	Easy	Hidden case	Success	15	0.3204 sec	54.4 KB
Testcase 6	Easy	Hidden case	Success	15	0.3187 sec	48.6 KB
Testcase 7	Easy	Hidden case	Success	15	0.3496 sec	53.5 KB
Testcase 8	Easy	Sample case	Success	0	0.0908 sec	29.4 KB

No Comments

PDF generated at: 8 Apr 2024 03:05:22 UTC