



Full Name: Yashavantha Kumara H O

Email: yashuuu930@gmail.com

Test Name: Mock Test

Taken On: 13 Aug 2025 07:23:46 IST

Time Taken: 24 min 52 sec/ 40 min

Invited by: Ankush

Invited on: 13 Aug 2025 07:22:48 IST

Skills Score:

Tags Score:

- Algorithms 160/195
- Constructive Algorithms 90/90
- Core CS 160/195
- Easy 70/105
- Greedy Algorithms 90/90
- Medium 90/90
- Problem Solving 160/195
- Search 70/105
- Sorting 70/105
- problem-solving 160/195

82.1%

160/195

scored in **Mock Test** in 24 min
52 sec on 13 Aug 2025 07:23:46
IST

Recruiter/Team Comments:

No Comments.

Plagiarism flagged

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

	Question Description	Time Taken	Score	Status
Q1	Find the Median > Coding	7 min 10 sec	70/ 105	✓
Q2	Flipping the Matrix > Coding	17 min 23 sec	90/ 90	!

QUESTION 1

Correct Answer

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: **C**

```
1 #include<stdio.h>
2 int main()
3 {
4     int n;
5     scanf("%d", &n);
6     int a[n];
7     for(int i=0; i<n; i++)
8     {
9         scanf("%d", &a[i]);
10    }
11
12    for(int i=0; i<n-1; i++){
13        int mid=i;
14        for(int j=i+1; j<n; j++){
15            if(a[j]<a[mid]){
```

```

16         mid=j;
17     }
18 }
19 int temp=a[i];
20 a[i]=a[mid];
21 a[mid]=temp;
22 }
23 printf("%d\n", a[n/2]);
24 return 0;
25 }

```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 1	Easy	Sample case	✔ Success	0	0.0065 sec	7.25 KB
Testcase 2	Easy	Hidden case	✔ Success	35	0.0509 sec	7.38 KB
Testcase 3	Easy	Hidden case	✔ Success	35	0.0391 sec	7.25 KB
Testcase 4	Easy	Hidden case	✘ Terminated due to timeout	0	2.0033 sec	6.88 KB

No Comments

QUESTION 2



Needs Review

Score 90

Flipping the Matrix > Coding

Algorithms

Medium

Greedy Algorithms

Constructive Algorithms

problem-solving

Core CS

Problem Solving

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

$matrix = [[1, 2], [3, 4]]$

```

1 2
3 4

```

It is 2×2 and we want to maximize the top left quadrant, a 1×1 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 `flippingMatrix` function in the editor below.

flippingMatrix has the following parameters:

- *int* *matrix*[2*n*][2*n*]: 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

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

Sample Input

STDIN	Function
-----	-----
1	q = 1
2	n = 2
112 42 83 119	matrix = [[112, 42, 83, 119], [56, 125, 56, 49], \
56 125 56 49	[15, 78, 101, 43], [62, 98, 114, 108]]
15 78 101 43	
62 98 114 108	

Sample Output

414

Explanation

Start out with the following $2n \times 2n$ matrix:

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

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] → [114, 101, 56, 83]), resulting in the matrix:

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

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

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

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

CANDIDATE ANSWER

Language used: C

```
1 #include<stdio.h>
2 int main(){
3     int q;
4     scanf("%d", &q);
5     while(q--){
6         int n;
7         scanf("%d", &n);
8         int S= n*2;
9         int mat[S][S];
10
11         for(int i=0; i<S; i++)
12         {
13             for(int j=0; j<S; j++){
14                 scanf("%d", &mat[i][j]);
15             }
16         }
17         long sum=0;
18         for (int i =0;i<n; i++ )
19         {
20             for(int j=0; j<n; j++){
21                 int a= mat[i][j];
22                 int b= mat[i][S-j-1];
23                 int c= mat[S-i-1][j];
24                 int d= mat[S-i-1][S-j-1];
25
26                 int max =a;
27                 if (b>max) max=b;
28                 if (c>max) max=c;
29                 if (d>max) max=d;
30                 sum +=max;
31             }
32         }
33         printf("%ld\n",sum);
34     }
35     return 0;
36 }
37 }
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 1	Easy	Sample case	✔ Success	0	0.0099 sec	7.5 KB
Testcase 2	Easy	Hidden case	✔ Success	15	0.0343 sec	7.63 KB
Testcase 3	Easy	Hidden case	✔ Success	15	0.052 sec	7.5 KB
Testcase 4	Easy	Hidden case	✔ Success	15	0.023 sec	7.38 KB
Testcase 5	Easy	Hidden case	✔ Success	15	0.0392 sec	7.75 KB
Testcase 6	Easy	Hidden case	✔ Success	15	0.0384 sec	7.63 KB
Testcase 7	Easy	Hidden case	✔ Success	15	0.0451 sec	7.5 KB
Testcase 8	Easy	Sample case	✔ Success	0	0.0079 sec	7.25 KB

No Comments