

#### Mock Test > joelvarghese356@gmail.com

Full Name: Joel Varghese

Email: joelvarghese356@gmail.com

Test Name: Mock Test

 Taken On:
 28 Mar 2024 09:52:28 IST

Time Taken: 14 min 43 sec/ 24 min

Linkedin: http://www.linkedin.com/in/joel-varghese-

54b51923a

+916376422354

Invited by: Ankush

Invited on: 28 Mar 2024 09:51:48 IST

Skills Score:

**Contact Number:** 

Tags Score: Algorithms 90/90

Constructive Algorithms 90/90

Core CS 90/90

Greedy Algorithms 90/90

Medium 90/90

Problem Solving 90/90 problem-solving 90/90

**100%** 90/90

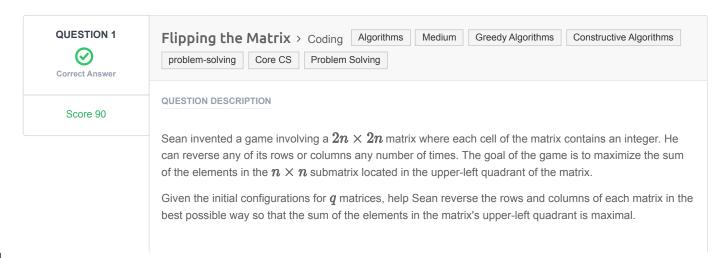
scored in **Mock Test** in 14 min 43 sec on 28 Mar 2024 09:52:28

IST

#### **Recruiter/Team Comments:**

No Comments.

Question Description	Time Taken	Score	Status
Q1 Flipping the Matrix > Coding	14 min 9 sec	90/ 90	<b>Ø</b>



# Example matrix = [[1, 2], [3, 4]]

```
1 2
3 4
```

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 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

- $1 \le q \le 16$
- $1 \le n \le 128$
- $ullet 0 \leq matrix[i][j] \leq 4096$ , 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:

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

3. Reverse row 0 ([112, 42, 114, 119] ightarrow [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: PyPy3

```
2 #
3 # Complete the 'flippingMatrix' function below.
4 #
5 # The function is expected to return an INTEGER.
6 # The function accepts 2D_INTEGER_ARRAY matrix as parameter.
7 #
8
9 def flippingMatrix(matrix):
     # Write your code here
     rows = len(matrix)
     col = len(matrix[0])
      sums = 0
     for i in range(0,rows//2):
       for j in range (0, col//2):
             sums += max(matrix[i][j], matrix[i][col-j-1], matrix[rows-1-i]
17 [j], matrix[rows-1-i][col-j-1])
     return sums
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 1	Easy	Sample case	Success	0	0.2624 sec	71.8 KB
Testcase 2	Easy	Hidden case	Success	15	0.3957 sec	75.7 KB
Testcase 3	Easy	Hidden case	Success	15	0.4358 sec	76.7 KB
Testcase 4	Easy	Hidden case	Success	15	0.3883 sec	75.8 KB
Testcase 5	Easy	Hidden case	Success	15	0.4039 sec	75.9 KB
Testcase 6	Easy	Hidden case	Success	15	0.434 sec	76.2 KB
Testcase 7	Easy	Hidden case	Success	15	0.4194 sec	76.5 KB
Testcase 8	Easy	Sample case	Success	0	0.2788 sec	71.6 KB

PDF generated at: 28 Mar 2024 04:38:39 UTC