

Mock Test > alenkuriakose.dev@gmail.com

Full Name:

Alen Kuriakose

Email:

alenkuriakose.dev@gmail.com

Test Name:

Mock Test

Taken On:

17 Sep 2024 22:31:30 IST

Time Taken:

23 min 56 sec/ 24 min

Invited by:

Ankush

Invited on:

17 Sep 2024 22:31:19 IST

Skills Score:

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%

scored in **Mock Test** in 23 min 56 sec on 17 Sep 2024 22:31:30 IST

Recruiter/Team Comments:

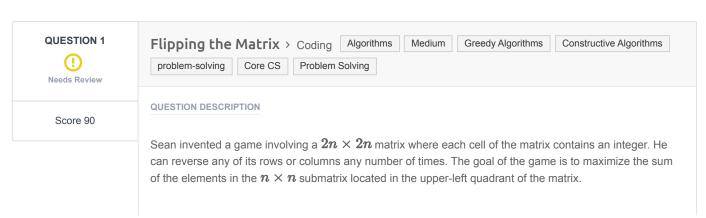
No Comments.

Plagiarism flagged

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

Question Description Time Taken Score Status

Q1 Flipping the Matrix > Coding 23 min 3 sec 90/90 1



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]
```

```
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[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 $2n \times 2n$ 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

The candidate did not manually submit any code. The last compiled version has been auto-submitted and the score you see below is for the auto-submitted version.

Language used: JavaScript (Node.js)

```
2 /*
   * Complete the 'flippingMatrix' function below.
   * The function is expected to return an INTEGER.
    * The function accepts 2D INTEGER ARRAY matrix as parameter.
   */
   function flippingMatrix(matrix) {
      // Write your code here
       const n=matrix.length/2
      let max=0;
      let total=0;
       for (let i=0; i<n; i++) {
           for (let j=0; j< n; j++) {
               max=Number.MIN VALUE;
               max=Math.max(matrix[i][j],max)
               max=Math.max(matrix[i][2*n -j -1],max)
               max=Math.max(matrix[2*n-i-1][j], max)
               max=Math.max(matrix[2*n-i-1][2*n-j-1],max)
               total+=max
           }
       return total
27 }
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 1	Easy	Sample case	Success	0	0.0445 sec	38.9 KB
Testcase 2	Easy	Hidden case	Success	15	0.0883 sec	53.4 KB
Testcase 3	Easy	Hidden case	Success	15	0.1188 sec	54.4 KB
Testcase 4	Easy	Hidden case	Success	15	0.0825 sec	51.3 KB
Testcase 5	Easy	Hidden case	Success	15	0.1155 sec	53.7 KB
Testcase 6	Easy	Hidden case	Success	15	0.1004 sec	54.4 KB
Testcase 7	Easy	Hidden case	Success	15	0.1061 sec	54.9 KB
Testcase 8	Easy	Sample case	Success	0	0.042 sec	38.8 KB
No Comments						

PDF generated at: 17 Sep 2024 17:27:20 UTC