



Full Name: Đào Nguyễn Huy Hoàng

Email: hoang.dnh210380@sis.hust.edu.vn

Test Name: Mock Test

Taken On: 4 Apr 2023 05:39:21 IST

Time Taken: 22 min 2 sec/ 30 min

Invited by: Ankush

Invited on: 4 Apr 2023 05:39:11 IST

Skills Score:

Tags Score:

100%

90/90

scored in **Mock Test** in 22 min 2 sec on 4 Apr 2023 05:39:21 IST

- 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

Recruiter/Team Comments:

No Comments.

	Question Description	Time Taken	Score	Status
Q1	Flipping the Matrix > Coding	21 min 53 sec	90/ 90	✓

QUESTION 1

✓

Correct Answer

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[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

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

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

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

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

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

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



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
3  int max(int a,int b){
4      if (a>b) return a;
5      return b;
6  }
7  int main(){
8      int s[1000][1000],n,q;
9      scanf("%d",&q);
10     for(int i=1;i<=q;i++)
11     {
12         scanf("%d",&n);
13         for(int i=1;i<=2*n;i++)
14         for(int j=1;j<=2*n;j++)
15             scanf("%d",&s[i][j]);
16         int ans = 0;
17         for(int i=1;i<=n;i++)
18             for(int j=1;j<=n;j++)
19             {
20                 int i2 = n + n + 1 - i;
21                 int j2 = n + n + 1 - j;
22                 int mx = max(max(max(s[i][j], s[i][j2]), s[i2][j]), s[i2][j2]);
23
24                 ans += mx;
25             }
26         printf("%d\n", ans);
27     }
28 }
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 1	Easy	Sample case	 Success	0	0.0226 sec	7.45 KB
Testcase 2	Easy	Hidden case	 Success	15	0.1022 sec	8.3 KB

Testcase 3	Easy	Hidden case	✔ Success	15	0.0886 sec	8.28 KB
Testcase 4	Easy	Hidden case	✔ Success	15	0.0541 sec	8.5 KB
Testcase 5	Easy	Hidden case	✔ Success	15	0.1103 sec	8.2 KB
Testcase 6	Easy	Hidden case	✔ Success	15	0.0828 sec	8.37 KB
Testcase 7	Easy	Hidden case	✔ Success	15	0.0981 sec	8.48 KB
Testcase 8	Easy	Sample case	✔ Success	0	0.0459 sec	7.37 KB

No Comments

PDF generated at: 4 Apr 2023 00:33:16 UTC