

Rotate Image by 90 degree



Rotate Image 90° $n \times n$

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

\rightarrow

13	9	5	1
14	10	6	2
15	11	7	3
16	12	8	4

✓

108 of 136



Handwritten notes on a digital blackboard illustrating a matrix transformation.

Initial Matrix (Left):

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

The first two columns (1, 5, 9, 13 and 2, 6, 10, 14) are circled in red. A red arrow points from the bottom of these circles to the text 2x2.

Transformation: A red arrow with a '2' above it points from the initial matrix to the final matrix.

Final Matrix (Right):

13	9	5	1
14	10	6	2
15	11	7	3
16	12	8	4

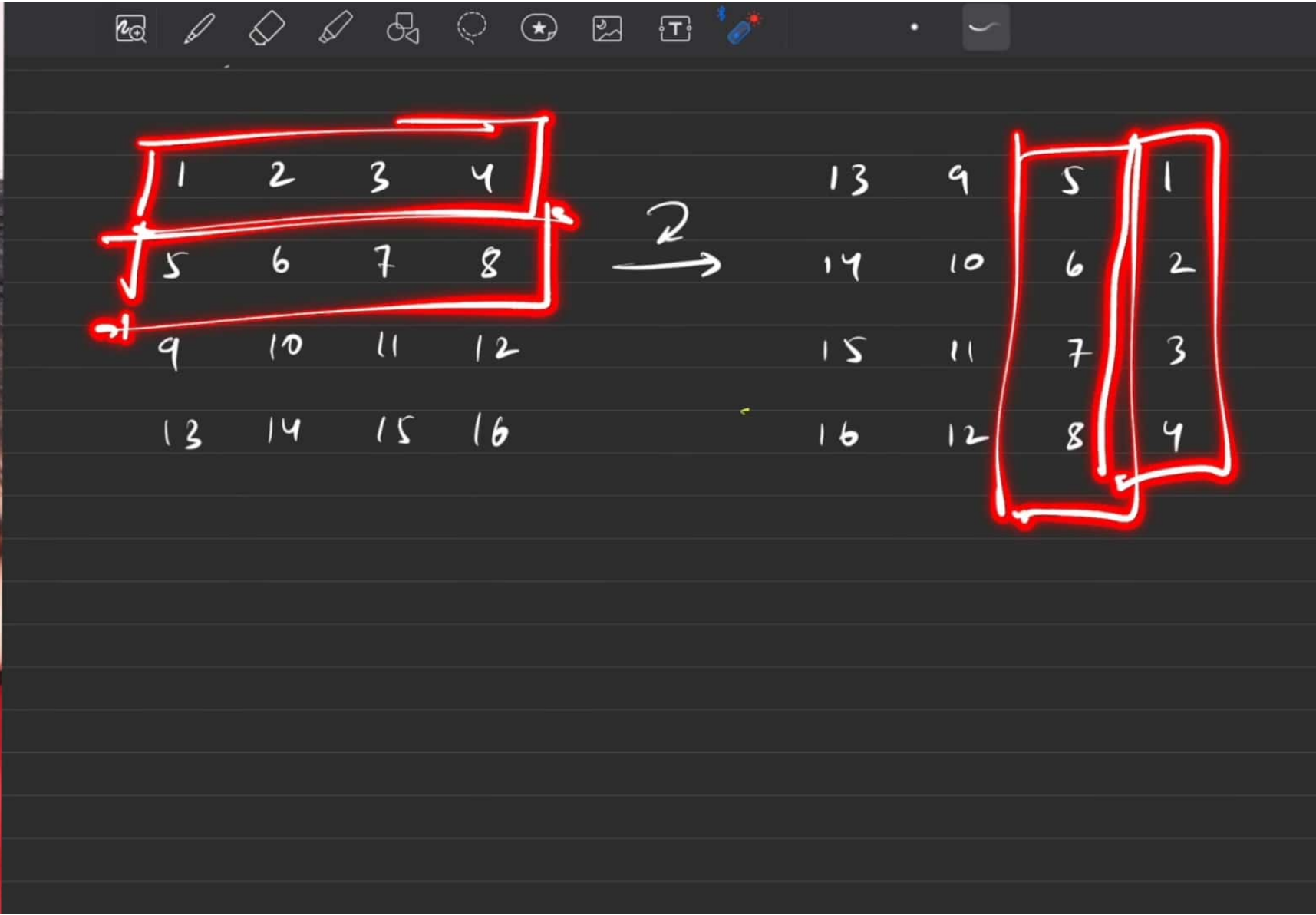
The final matrix is enclosed in a large red bracket on the right side, with the text n x n written below it.

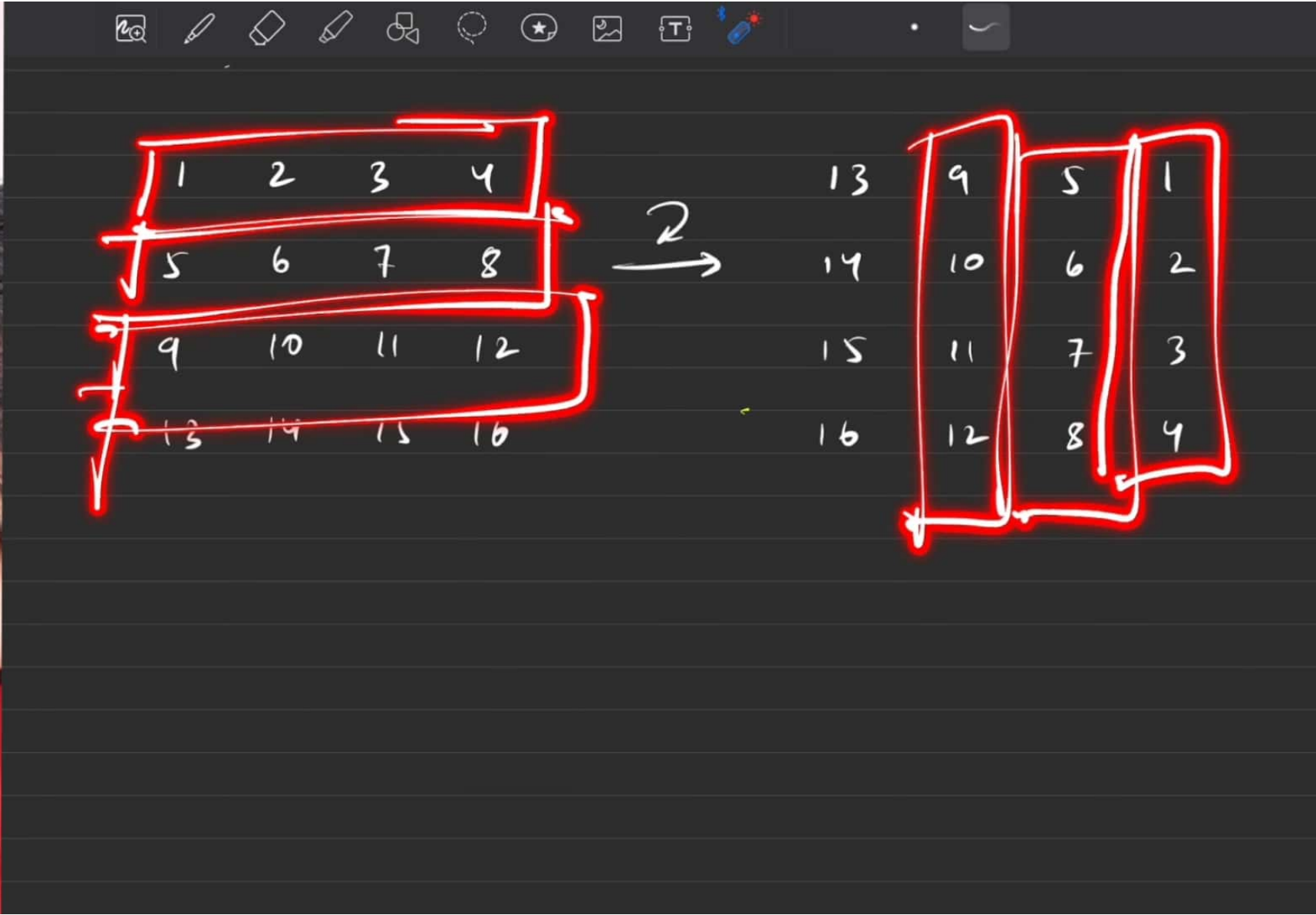
Header: The word "Brute" is written in red and enclosed in a red box at the top right.

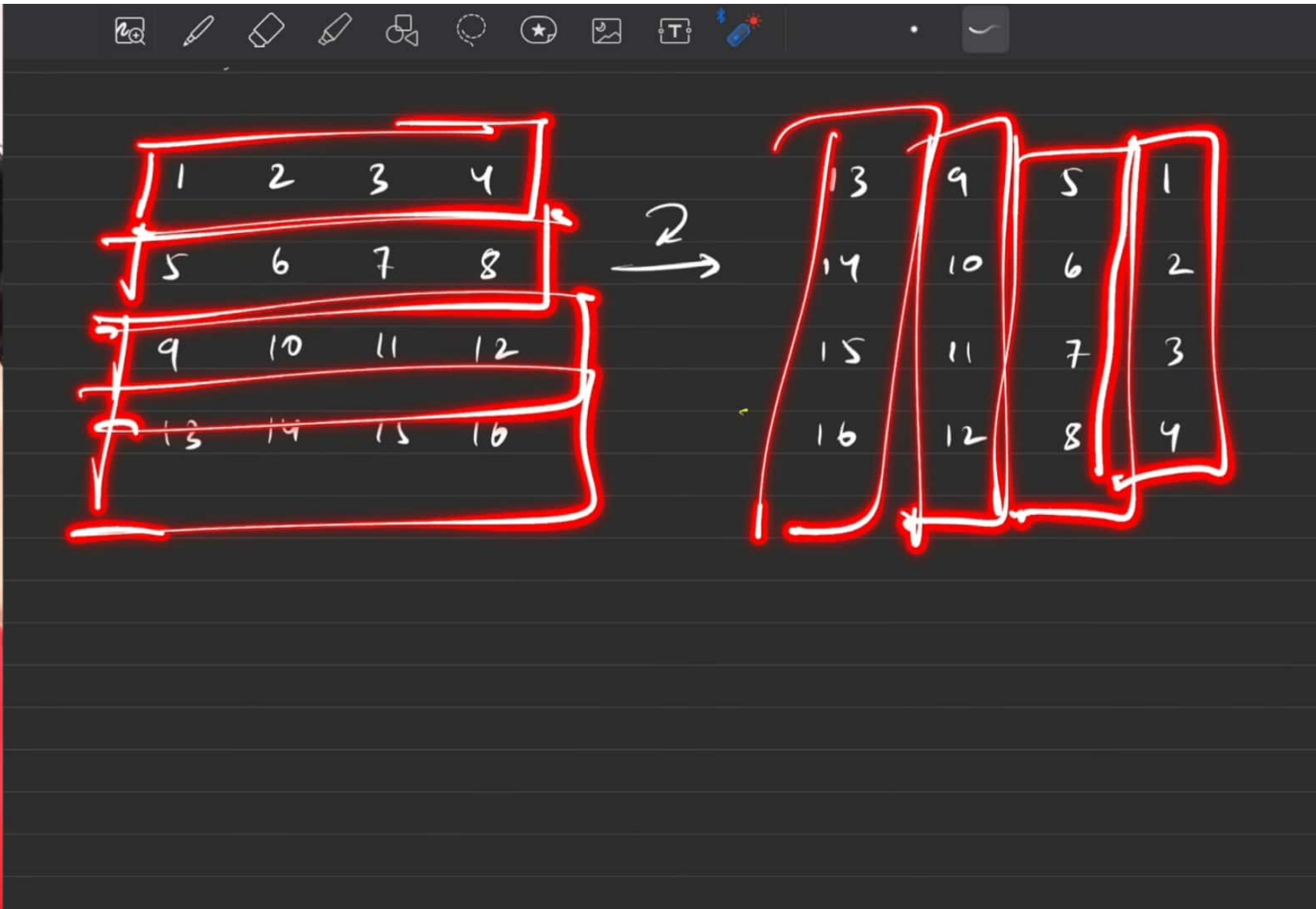


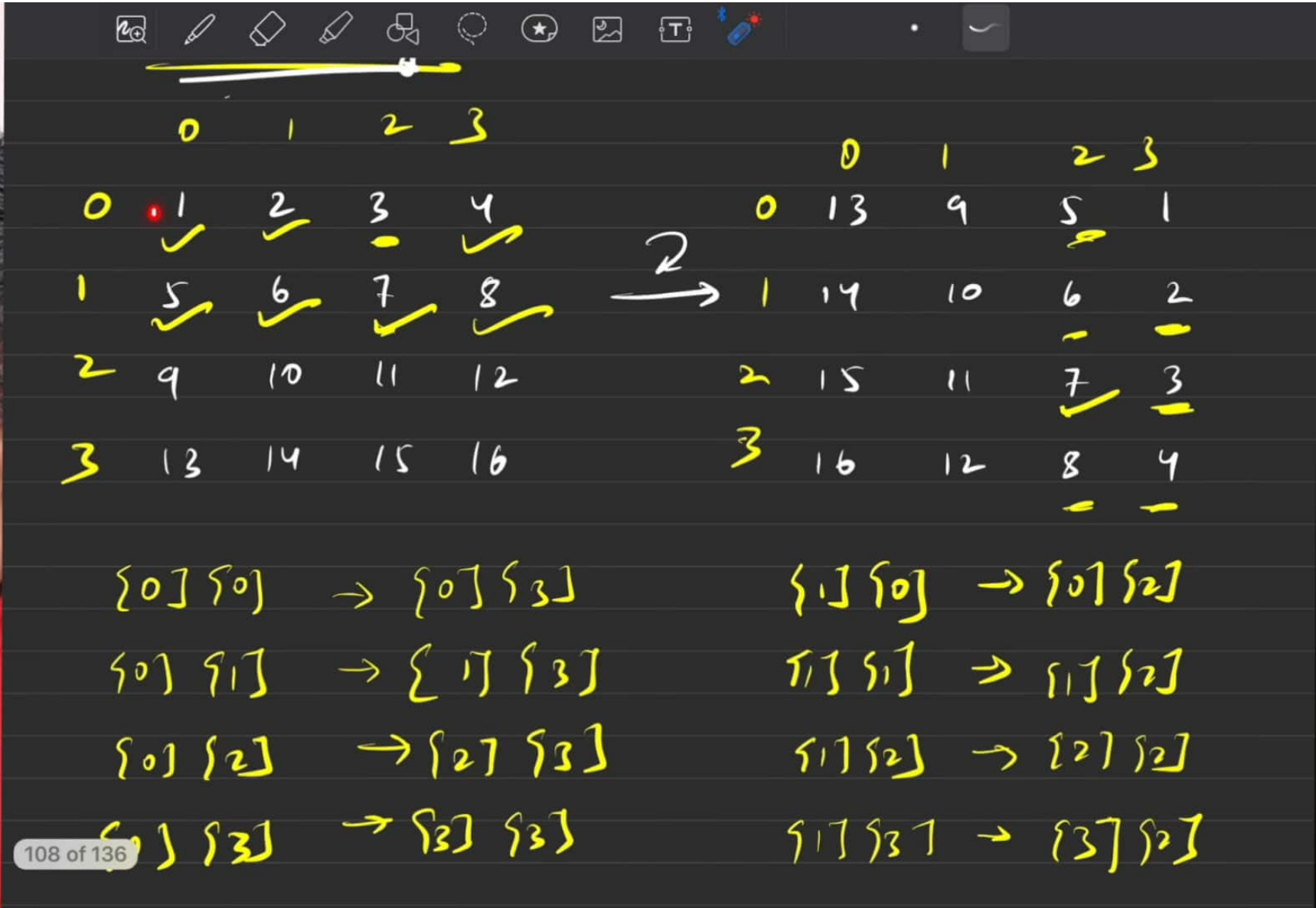
A digital whiteboard interface showing a 4x4 grid of numbers. The grid is divided into two sections by a large arrow pointing from left to right. The left section contains numbers 1 through 16, and the right section contains numbers 13 through 16. The numbers are arranged in a 4x4 grid. The first row of the left section is 1, 2, 3, 4, and the first row of the right section is 13, 9, 5. The second row of the left section is 5, 6, 7, 8, and the second row of the right section is 14, 10, 6. The third row of the left section is 9, 10, 11, 12, and the third row of the right section is 15, 11, 7. The fourth row of the left section is 13, 14, 15, 16, and the fourth row of the right section is 16, 12, 8. The numbers 1, 2, 3, 4 in the top row of the left section are enclosed in a red box. The numbers 1, 2, 3, 4 in the first column of the right section are enclosed in a red box. A large arrow points from the left section to the right section.

1	2	3	4		13	9	5
5	6	7	8	→	14	10	6
9	10	11	12		15	11	7
13	14	15	16		16	12	8











Handwritten notes on a digital drawing board, showing two sets of numbers and their corresponding interval mappings.

Left Set:

	9	10	11	12
2				
3	13	14	15	16

Interval mappings for the left set:

- $\{0] \rightarrow [0]$ (with i and j above)
- $[0] \rightarrow [1]$
- $[0] \rightarrow [2]$
- $[0] \rightarrow [3]$

Right Set:

	15	11	7	3
2				
3	16	12	8	4

Interval mappings for the right set:

- $[1] \rightarrow [0]$ (with j above)
- $[1] \rightarrow [1]$
- $[1] \rightarrow [2]$
- $[1] \rightarrow [3]$



$n=4$

$0 \rightarrow \underline{\underline{3}}$

$i \rightarrow (n-1)$

$0 \rightarrow (n-1) - 0$

$\underline{\underline{n-1 = 3}}$

\downarrow

1

$(i) \rightarrow (n-1) - i$

$1 \rightarrow 2$

$\underline{\underline{i}} \rightarrow (n-1) - i$

$1 \rightarrow (n-1) - 1$

$1 \rightarrow \underline{\underline{n-2}}$

\downarrow

2



3

13

14

15

16

i

j

$(n-1)-i$

$\{0\} \{0\}$

\rightarrow

$\{0\} \{3\}$

$\{0\} \{1\}$

\rightarrow

$\{1\} \{3\}$

$\{0\} \{2\}$

\rightarrow

$\{2\} \{3\}$

$\{0\} \{3\}$

\rightarrow

$\{3\} \{3\}$

$n=4$

3

16

12

8

4

j

i

$(n-1)-j$

$\{1\} \{0\}$

\rightarrow

$\{0\} \{2\}$

$\{1\} \{1\}$

\rightarrow

$\{1\} \{2\}$

$\{1\} \{2\}$

\rightarrow

$\{2\} \{2\}$

$\{1\} \{3\}$

\rightarrow

$\{3\} \{2\}$

$0 \rightarrow 3$

$i \rightarrow (n-1)$

$0 \rightarrow (n-1)-0$

$1 \rightarrow 2$

$j \rightarrow (n-1)-j$

$1 \rightarrow (n-1)-1$



Handwritten code and complexity analysis on a digital notepad:

```
ans[n][n]
for (i = 0 -> n)
{
    for (j = 0 -> n)
    {
        ans[j][n-1-i] = mat[n-1-i][j]
    }
}
```

Complexity Analysis:

- TC $\rightarrow O(n^2)$
- SL $\rightarrow O(n^2)$

A blue arrow points from the underlined assignment statement in the code to the space complexity analysis.



Rotate Image

↻ 90°

n=4

	0	1	2	3
0	1	2	3	4
1	5	6	7	8
2	9	10	11	12
3	13	14	15	16



	0	1	2	3
0	13	9	5	1
1	14	10	6	2
2	15	11	7	3
3	16	12	8	4

In-place

$O(n^2)$



Rotate Image $\curvearrowright 90^\circ$ $n=4$

	0	1	2	3
0	1	2	3	4
1	5	6	7	8
2	9	10	11	12
3	13	14	15	16

\rightarrow

	0	1	2	3
0	13	9	5	1
1	14	10	6	2
2	15	11	7	3
3	16	12	8	4



Rotate Image $\curvearrowright 90^\circ$ $n=4$

	0	2	3	
0	1	2	3	4
1	5	6	7	8
2	9	10	11	12
3	13	14	15	16

\rightarrow

	0	1	2	3
0	13	9	5	1
1	14	10	6	2
2	15	11	7	3
3	16	12	8	4



Handwritten notes on a digital screen showing a 4x5 grid of numbers and a transformation process.

Initial Grid:

0	1	2	3	4
0	1	2	3	4
1	5	6	7	8
2	9	10	11	12
3	13	14	15	16

Transformation: A large arrow labeled '2' points from the initial grid to the transformed grid.

Transformed Grid:

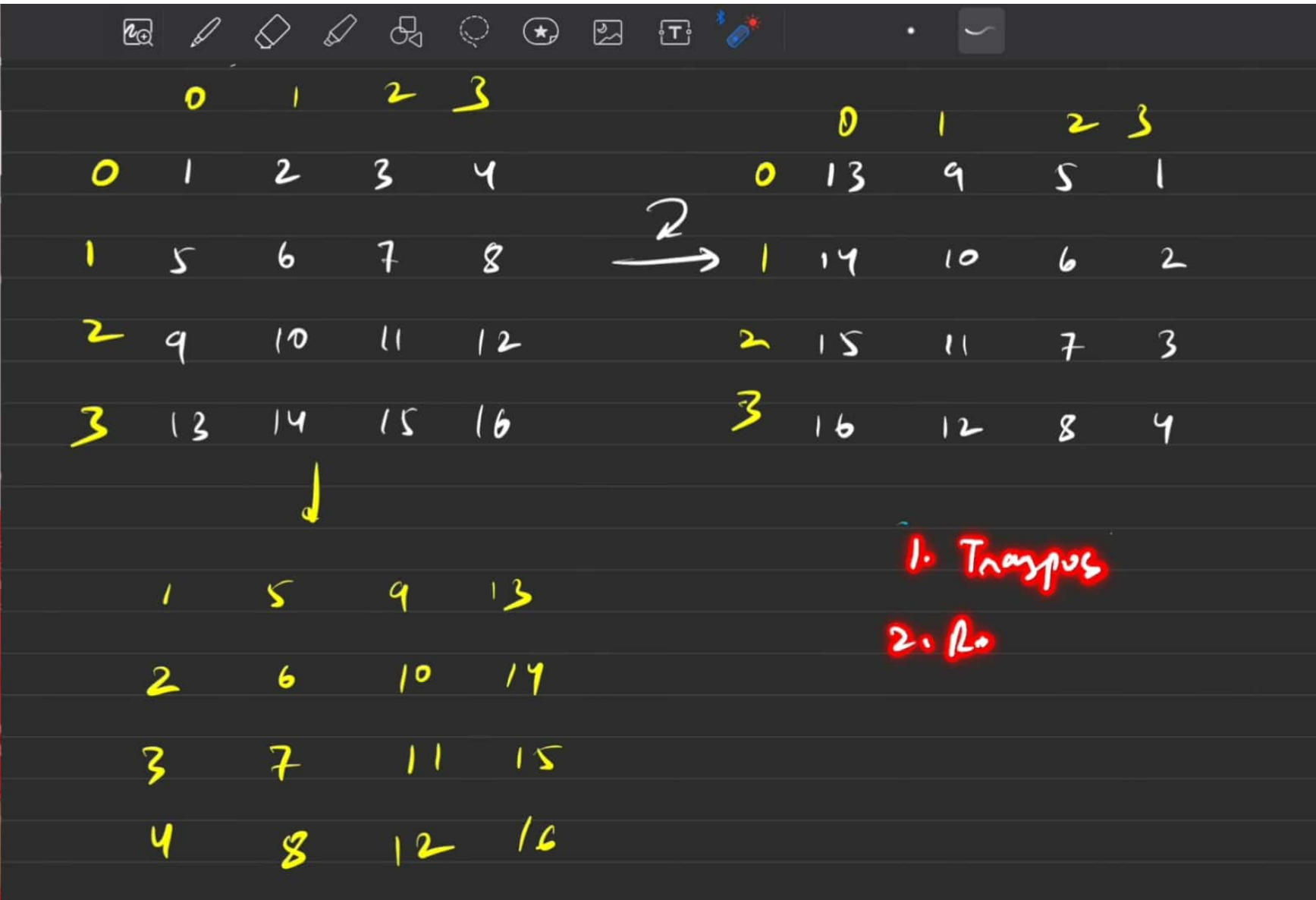
0	1	2	3	
0	13	9	5	1
1	14	10	6	2
2	15	11	7	3
3	16	12	8	4

Transpose: A downward arrow labeled 'Transpose' points to the final grid.

Final Grid:

1	5	9	13
2	6	10	14
3	7	11	15
4	8	12	16

Annotation: A curved arrow points from the final grid to the text "reverse every row".





0 1 2 3						0 1 2 3				
0	1	2	3	4	2 →	0	13	9	5	1
1	5	6	7	8		1	14	10	6	2
2	9	10	11	12		2	15	11	7	3
3	13	14	15	16		3	16	12	8	4
↓										
	1	5	9	13						
2		6	10	14						
3		7	11	15						
4		8	12	16						



	0	1	2	3		0	1	2	3
0	1	2	3	4	$\xrightarrow{2}$	0	13	9	5
1	5	6	7	8		1	14	10	6
2	9	10	11	12		2	15	11	7
3	13	14	15	16		3	16	12	8
	1	5	9	13					
2	6	10	14						
3	7	11	15						
4	8	12	16						



Handwritten notes on a digital notepad showing a sequence of numbers and a transformation.

Initial Sequence (Left):

	0	1	2	3
0	1	2	3	4
1	5	6	7	8
2	9	10	11	12
3	13	14	15	16

A blue diagonal line is drawn from the top-left (0,0) to the bottom-right (3,3). A yellow arrow points down from the cell (2,2) to the cell (0,1).

Transformed Sequence (Right):

	0	1	2	3
0	13	9	5	1
1	14	10	6	2
2	15	11	7	3
3	16	12	8	4

A curved arrow points from the initial sequence to the transformed sequence.

Recurrence Relations (Bottom Right):

$$\{0\} \{1\} \rightarrow \{1\} \{0\}$$
$$\{0\} \{2\} \rightarrow \{2\} \{0\}$$
$$\{0\} \{3\} \rightarrow \{3\} \{0\}$$



Handwritten notes on a digital whiteboard illustrating a sorting process, likely Merge Sort, using a 4x4 grid of numbers.

Initial Grid (Left):

	0	1	2	3
0	1	2	3	4
1	5	6	7	8
2	9	10	11	12
3	13	14	15	16

Annotations: A blue diagonal line crosses the grid from (0,1) to (3,4). Elements 7, 8, 12, and 15 are boxed. A yellow arrow points down from the box around 15.

Transformed Grid (Right):

	0	1	2	3
0	13	9	5	1
1	14	10	6	2
2	15	11	7	3
3	16	12	8	4

Final Sorted Grid (Bottom):

1	5	9	13
2	6	10	14
3	7	11	15
4	8	12	16

Recursion/Logic Notes (Right):

- $\{0\} \{1\} \rightarrow \{1\} \{0\}$
- $\{0\} \{2\} \rightarrow \{2\} \{0\}$
- $\{0\} \{3\} \rightarrow \{3\} \{0\}$
- $\{1\} \{2\} \rightarrow \{2\} \{1\}$
- $\{1\} \{3\} \rightarrow \{3\} \{1\}$



Handwritten notes on a digital blackboard:

Top toolbar: Eraser, Pencil, Highlighter, Lasso, Circle, Star, Image, Text, Stamp, and window controls.

Diagram illustrating the swap operation in an array:

$\rightarrow [2] [3] \rightarrow [3] [2]$

Diagram showing the range of indices for the first element:

$i \rightarrow (1 \text{ to } 3)$
 $\quad \quad \quad \underline{i+1} \quad n-1$

Diagram showing the range of indices for the second element:

$j \rightarrow (2 \text{ to } 3)$
 $\quad \quad \quad \downarrow \quad \quad \downarrow$
 $\quad \quad \quad i+1 \quad (n-1)$

Code snippet for the swap operation:

```
for (i = 0 → n-2)
{
    for (j = i+1 → n-1)
    {
        swap (a[i], a[j]);
    }
}
```



Handwritten notes on a digital blackboard illustrating a sequence of operations and a resulting table.

Top Left Table (4x4):

0	1	2	3	4
1	5	6	7	8
2	9	10	11	12
3	13	14	15	16

Annotations: A blue diagonal line runs from (0,1) to (3,4). A yellow line connects (0,2) to (2,4). A red arrow points from (2,3) to (0,0). A yellow circle highlights (2,3). A blue circle highlights (1,3). A blue box highlights (3,3). A red arrow points from (3,3) to the bottom table.

Top Right Table (4x4):

0	13	9	5	1
1	14	10	6	2
2	15	11	7	3
3	16	12	8	4


Bottom Table (4x4):

1	5	9	13
2	6	10	14
3	7	11	15
4	8	12	16


Right Side Operations:

- $\{0\} \{1\} \rightarrow \{1\} \{0\}$
- $\{0\} \{2\} \rightarrow \{2\} \{0\}$
- $\{0\} \{3\} \rightarrow \{3\} \{0\}$
- $\{1\} \{2\} \rightarrow \{2\} \{1\}$
- $\{1\} \{3\} \rightarrow \{3\} \{1\}$
- $\{2\} \{3\} \rightarrow \{3\} \{2\}$



codestudio
POWERED BY CODING NINJAS

[Guided Paths](#) [Contests](#) [Interview Prep](#) [Practice](#) [Resources](#)



 [Problem of the day](#)

[Topics](#) [Problem](#) [Submissions](#) [Solution](#) [New](#)

Current Submission [Report](#)





Correct Answer Test Cases EXP: 11/11 Penalty 0% C++

few secs ago 40/40




Did you find these test cases useful?  

Previous Submissions [All languages](#)

Status	Test cases	EXP	Penalty	Language
✓	11/11	+ 40 ⚡	0%	C++
✓	11/11	+ 40 ⚡	0%	C++

```
1 #include<bits/stdc++.h>
2 void rotateMatrix(vector<vector<int>> &mat){
3     int n = mat.size();
4     // transpose
5     // O(N / 2 * N / 2)
6     for(int i = 0; i<n-1; i++) {
7         for (int j = i + 1; j < n; j++) {
8             swap(mat[i][j], mat[j][i]);
9         }
10    }
11    // reverse
12    // O(N * N / 2)
13    for(int i = 0; i<n; i++) {
14        // row is mat[i]
15        reverse(mat[i].begin(), mat[i].end());
16    }
17 }
```

 View hints  Last saved on 5:17:40 AM  R

Merge Overlapping Intervals

Merge Overlapping Subintervals

$(1, 3)$ $(2, 6)$ $(8, 9)$ $(9, 11)$ $(8, 10)$ $(2, 4)$ $(15, 18)$ $(16, 17)$

$\{1, 6\}$ $\{8, 11\}$ $\{15, 18\}$



Merge Overlapping Subintervals

$(1, 3)$ $(2, 6)$ $(8, 9)$ $(9, 11)$ $(8, 10)$ $(2, 4)$ $(15, 18)$ $(16, 17)$

$\{1, 6\}$ $\{8, 11\}$ $\{15, 18\}$

$\overline{\overline{1 \quad 2 \quad 3 \quad 4 \quad 6}}$

$(1, 6)$

$\overline{\overline{8 \quad 9 \quad 10 \quad 11}}$

$\overline{\overline{15 \quad 16 \quad 17 \quad 18}}$

Merge Overlapping Subintervals

$(1, 3)$ $(2, 6)$ $(8, 9)$ $(9, 11)$ $(8, 10)$ $(2, 4)$ $(15, 18)$ $(16, 17)$

✓ sort

$(1, 3)$



Merge Overlapping Subintervals

(1, 3) Q₂ (4, 6) (8, 9) (9, 11) (8, 10) Q₂ (15, 18) (16, 17)

✓ sort

(1, 3)

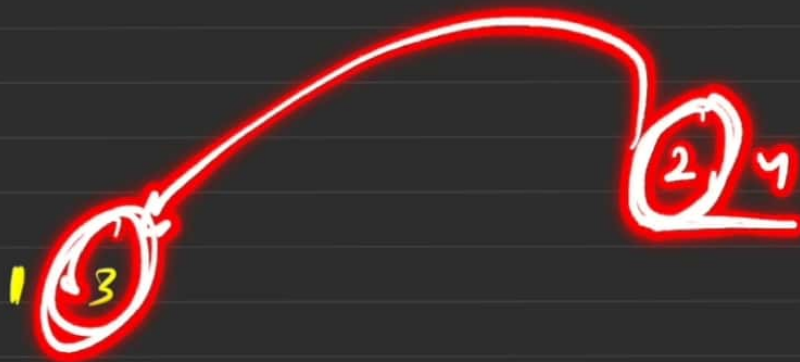


Merge Overlapping Subintervals

(1, 3) (2, 6) (8, 9) (9, 11) (8, 10) (2, 4) (15, 18) (16, 17)

✓ sort

(1, 3) (2, 4) (2, 6) (8, 9) (8, 10) (9, 11) (15, 18) (16, 17)





Merge Overlapping Subintervals

(1, 3) (2, 6) (8, 9) (9, 11) (8, 10) (2, 4) (15, 18) (16, 17)

✓ sort

(1, 3) (2, 4) (2, 6) (8, 9) (8, 10) (9, 11) (15, 18) (16, 17)



1, 4

Merge Overlapping Subintervals

$(1, 3)$ $(2, 6)$ $(8, 9)$ $(9, 11)$ $(8, 10)$ $(2, 4)$ $(15, 18)$ $(16, 17)$

✓ sort

$(1, 3)$ $(2, 4)$ $(2, 6)$ $(8, 9)$ $(8, 10)$ $(9, 11)$ $(15, 18)$ $(16, 17)$





Merge Overlapping Subintervals

$(1, 3)$ $(2, 6)$ $(8, 9)$ $(9, 11)$ $(8, 10)$ $(2, 4)$ $(15, 18)$ $(16, 17)$

sort

$(1, 3)$ $(2, 4)$ $(2, 6)$ $(8, 9)$ $(8, 10)$ $(9, 11)$ $(15, 18)$ $(16, 17)$



$(1, 6)$



Merge Overlapping Subintervals

$(1, 3)$ $(2, 6)$ $(8, 9)$ $(9, 11)$ $(8, 10)$ $(2, 4)$ $(15, 18)$ $(16, 17)$



$(1, 3)$ $(2, 4)$ $(2, 6)$ $(8, 9)$ $(8, 10)$ $(9, 11)$ $(15, 18)$ $(16, 17)$





Merge Overlapping Subintervals

(1, 3) (2, 6) (8, 9) (9, 11) (8, 10) (2, 4) (15, 18) (16, 17)

sort

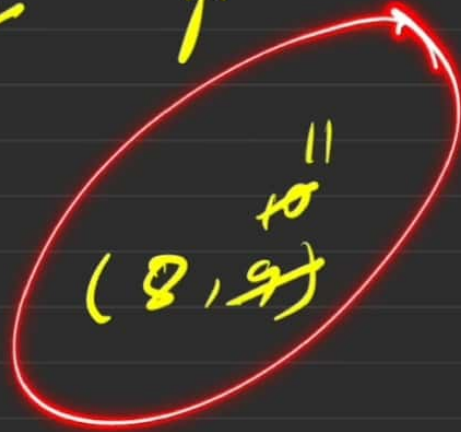
(1, 3) (2, 4) (2, 6) (8, 9) (8, 10) (9, 11) (15, 18) (16, 17)

↑ ↑ ↑ ↑

(1, 6)

(8, 9)
10
11

2
→





Merge Overlapping Subintervals

$(1, 3)$ $(2, 6)$ $(8, 9)$ $(9, 11)$ $(8, 10)$ $(2, 4)$ $(15, 18)$ $(16, 17)$

sort

$(1, 3)$ $(2, 4)$ $(2, 6)$ $(8, 9)$ $(8, 10)$ $(9, 11)$ $(15, 18)$ $(16, 17)$

↗ ↗ ↗ ↗ ↗ ↗ ↗ ↗

$[1, 6)$ $(8, 11)$ $(15, 18)$



codestudio

Guided PathsContestsInterview PrepPracticeResources

Problem of the day

TopicsProblemSubmissionsSolutionNew

Current Submission

Correct Answer

Test Cases EXP: 11/11

Penalty 0%

C++

few secs ago

80/80

Did you find these test cases useful?

Previous Submissions

All languages

Status	Test cases	EXP	Penalty	Language
✓	11/11	+ 80	0%	C++

```
1 #include<bits/stdc++.h>
2 vector<vector<int>> mergeOverlappingIntervals(vector<vector<int>
3     int n = arr.size();
4     sort(arr.begin(), arr.end());
5     vector<vector<int>> ans;
6     for(int i = 0; i<n; i++) {
7         int start = arr[i][0];
8         int end = arr[i][1];
9         if(!ans.empty() && end <= ans.back()[1]) {
10             continue;
11         }
12         for(int j = i+1; j<n; j++) {
13             if(arr[j][0] <= end) {
14                 end = max(end, arr[j][1]);
15             }
16             else {
17                 break;
18             }
19         }
20         ans.push_back({start, end});
21     }
22
23     return ans;
24 }
```

< Prev

Next >

View hints

✓ Last saved on 2:05:28 AM



(1, 3) (2, 6) (8, 9) (9, 11) (8, 10) (2, 4) (15, 18) (16, 17)

sorted

(1, 3) (2, 4) (2, 6) (8, 9) (8, 10) (9, 11) (15, 18) (16, 17)

↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑

start = 1
end = 3



Handwritten notes on a digital whiteboard interface. The interface includes a toolbar at the top with various drawing tools (pen, eraser, highlighter, selection tools, etc.) and a dark background with horizontal lines.

Handwritten text in blue ink:

$(1, 2) \quad (3, 4) \quad (6, 2)$

Handwritten text in yellow ink:

$\frac{n \log w}{o(w)} + \frac{o(2n)}{o(w)}$



Merge Overlapping Subintervals

$(1, 3)$ $(2, 6)$ $(8, 9)$ $(9, 11)$ $(8, 10)$ $(2, 4)$ $(15, 18)$ $(16, 17)$

✓ sort

$(1, 3)$ $(2, 4)$ $(2, 6)$ $(8, 9)$ $(8, 10)$ $(9, 11)$ $(15, 18)$ $(16, 17)$

↑ ↑ ↑ ↑

$(1, 6)$ $(8, 9)$



Merge Overlapping Subintervals

$(1, 3)$ $(2, 6)$ $(8, 9)$ $(9, 11)$ $(8, 10)$ $(2, 4)$ $(15, 18)$ $(16, 17)$

✓ sort

$(1, 3)$ $(2, 4)$ $(2, 6)$ $(8, 9)$ $(8, 10)$ $(9, 11)$ $(15, 18)$ $(16, 17)$
↑ ↑ ↑ ↑ ↑ ↑ ↑

$(1, 6)$ $(8, 11)$



Merge Overlapping Subintervals

$(1, 3)$ $(2, 6)$ $(8, 9)$ $(9, 11)$ $(8, 10)$ $(2, 4)$ $(15, 18)$ $(16, 17)$

sort

$(1, 3)$ $(2, 4)$ $(2, 6)$ $(8, 9)$ $(8, 10)$ $(9, 11)$ $(15, 18)$ $(16, 17)$
↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑

$(1, 6)$ $(8, 11)$ $(15, 18)$



codestudio

Guided Paths Contests Interview Prep Practice Resources

Topics Problem Submissions Solution

Hey Champ, hold on we are checking... Report

1. Compilation check ✓

2. Large test cases check ✓

3. Optimal code test ○

Previous Submissions

All languages

Status	Test cases	EXP	Penalty	Language
✓	11/11	+ 80	0%	C++
✓	11/11	+ 80	0%	C++
-	-	0%	-	C++

```
1 #include<bits/stdc++.h>
2 vector<vector<int>> mergeOverlappingIntervals(vector<vector<int>
3     int n = arr.size();
4     sort(arr.begin(), arr.end());
5     vector<vector<int>> ans;
6     for(int i = 0; i < n; i++) {
7         if(ans.empty() || arr[i][0] > ans.back()[1]) {
8             ans.push_back(arr[i]);
9         }
10        else {
11            ans.back()[1] = max(ans.back()[1], arr[i][1]);
12        }
13    }
14    return ans;
15 }
```

Merge Overlapping Subintervals

(1, 3) (2, 6) (8, 9) (9, 11) (8, 10) (2, 4) (15, 18) (16, 17)

/ sort

(1, 3) (2, 4) (2, 6) (8, 9) (8, 10) (9, 11) (15, 18) (16, 17)

↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑

(1, 6) (8, 11) (15, 18)

< Prev Next >

View hints

✓



codestudio

Guided PathsContestsInterview PrepPracticeResources

Problem of the day

TopicsProblemSubmissionsSolutionNew

Current Submission

Report

Correct AnswerTest Cases EXP: 11/11Penalty 0% C++
few secs ago80/80

Did you find these test cases useful?

Previous Submissions

All languages

Status	Test cases	EXP	Penalty	Langue
✓	11/11	+ 80	0%	C++
✓	11/11	+ 80	0%	C++
✓	11/11	Already earned	0%	C++

```
1 #include<bits/stdc++.h>
2 vector<vector<int>>> mergeOverlappingIntervals(vector<vector<int>>> &arr){
3     int n = arr.size();
4     sort(arr.begin(), arr.end());
5     vector<vector<int>>> ans;
6     for(int i = 0; i<n; i++) {
7         if(ans.empty() || arr[i][0] > ans.back()[1]) {
8             ans.push_back(arr[i]);
9         }
10        else {
11            ans.back()[1] = max(ans.back()[1], arr[i][1]);
12        }
13    }
14    return ans;
15 }
```

(1,3) (2,4) (2,6) (8,9) (8,10) (9,11) (15,18) (16,17)

↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑

(1,6) (8,11) (15,18)

TL → No of M = O(N)

SL → O(N)

< PrevNext >

View hints✓