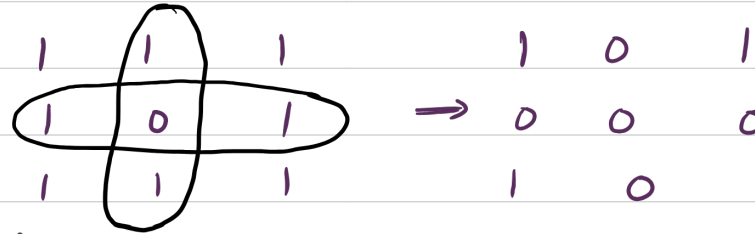
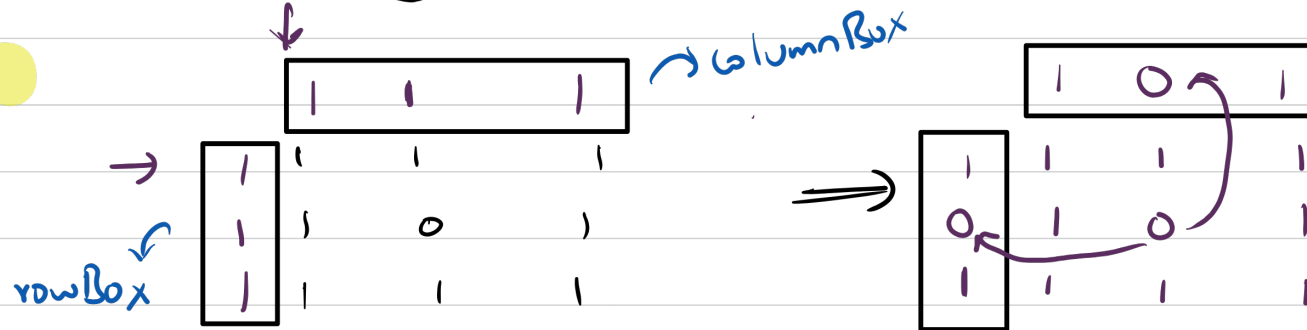


Set Matrix to zeroes

Given an $M \times N$ matrix, if an element is 0, Set entire row & Column values to 0



Intuition - ①



* Create a row Box & Col Box, Set to '1' initially

Step 1 * Traverse through the array and make rowBox value & ColBox value of particular index to zero, if you found matrix value as 0

Step 2 * Then re-iterate on matrix & check rowBox & ColBox, if any of them is zero, make the matrix value to 0

```
int rowBox [ ] = new int[col...size]
```

```
int colBox [ ] = new int[row...size]
```

```
for (int i=0; i<m; i++) Step 1
{
    for (int j=0; j<n; j++)
    {
        if (matrix[i][j] == 0)
        {
            rowBox[i] = 0;
            colBox[j] = 0;
        }
    }
}
```

```
for (int i=0; i<m; i++) Step 2
{
    for (int j=0; j<n; j++)
    {
        if (rowBox[i] == 0 || colBox[j] == 0)
        {
            matrix[i][j] = 0;
        }
    }
}
```

Time complexity:

$$O(n^2)$$

Space complexity:

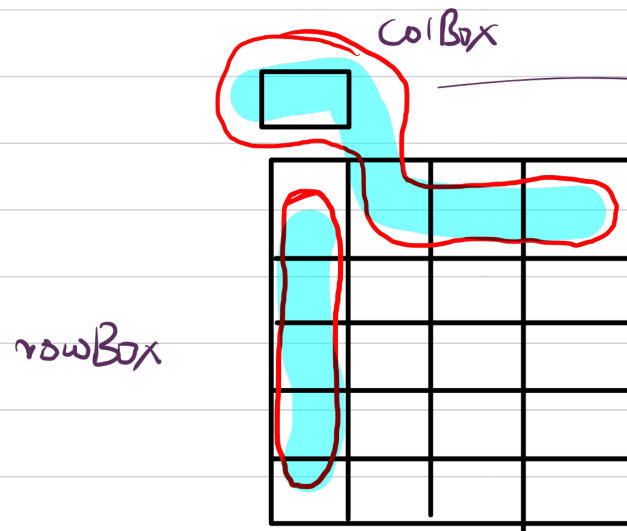
$$O(n) + O(m)$$

We can't reduce time complexity, But we can reduce Space complexity

Intuition

Instead of using separate arrays for rowBox & colBox

Assume \Rightarrow



Consider them as colBox & rowBox

rowBox \Rightarrow col[0]

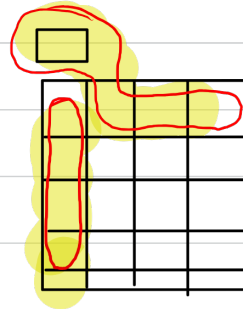
colBox \Rightarrow col[0, row[0]

from 1st index of column

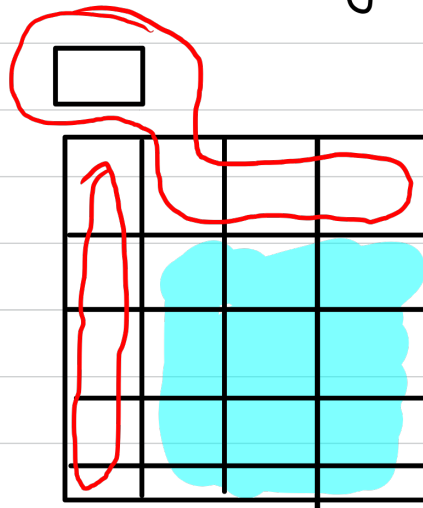
variable

* Traverse for (1,1) to (n,n) without include first row & first column

* Make rowBox, colBox as zeroes



* Iterate from (1,1) to (n,n) again and make them zeroes



* After that, Modify columnBox \Rightarrow means 0th row

↳ Because, 0th row is dependent on 0th col
if you modify 0th col first, then 0th row ans will change

* Then modify \Rightarrow 0th column

↳ 0th col is only dependent on another Box called (col0)

$$\underline{\underline{TC}}: O(n^2)$$

$$\underline{\underline{SC}}: O(1)$$

Dry Run

col = 1

1	1	1	1
1	0	1	1
1	1	0	1
0	1	1	1

col = 1

0	1	0	1	1
1	1	0	1	1
2	1	1	0	1
3	0	1	1	1
	0	1	2	3

i j
1 1

col = 1

0	1	0	1	1
1	1	0	1	1
2	1	1	0	1
3	0	1	1	1
	0	1	2	3

i j
3 0

col = 1

0	1	0	1	1
1	1	0	1	1
2	1	1	0	1
3	0	1	1	1
	0	1	2	3

i j
2 2

```
for (int i = 1; i < m; i++) {
    for (int j = 1; j < n; j++) {
        if (matrix[i][j] == 0) {
            // rowbox value setting to 0
            matrix[i][0] = 0;
            // colbox value setting to 0
            matrix[0][j] = 0;
        }
    }
}
```

col0 = 1

1	1	1	1
1	0	1	1
1	1	0	1
0	1	1	1

* if you have any zeroes in blue region, we update our rowBox & colBox

* What if,

↳ we have zeroes in rowBox & colBox itself

* So, if any index of rowBox have zero,

↳ it depends on col0 variable
update it to 0

* So, if any index of colBox have zero,

↳ it depends on row[0][0]
update it to 0

```
// Setting 0th col valueBox to zero
for (int i = 0; i < m; i++) {
    if (matrix[i][0] == 0) {
        col0 = 0;
    }
}
```

```
// Setting 0th row valueBox to zero
for (int i = 1; i < n; i++) {
    if (matrix[0][i] == 0) {
        matrix[0][0] = 0;
    }
}
```

col = 0

1	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1

Step 1 → done

Now Again traverse from $(1,1)$ to (m,n) and modify 1's to 0's
if condition Satisfies

then do 0^{th} row , then 0^{th} column

Check code for clarification