

28. Set Matrix Zeros

Given an $m \times n$ integer matrix `matrix`, if an element is `0`, set its entire row and column to `0`'s.

You must do it in place.

Example 1:

1	1	1
1	0	1
1	1	1



1	0	1
0	0	0
1	0	1

Input: `matrix = [[1,1,1],[1,0,1],[1,1,1]]`
Output: `[[1,0,1],[0,0,0],[1,0,1]]`

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

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

i) Brute force :

→ Iterate over the matrix

→ if we find '0' don't do anything

— if it's '0' then we mark the row & col as '0' .

BUT It's not correct because array will look different

and again on next iteration for the next '0' you
mark again the row and column will be '0'.

and again on next iteration for the '0' - you mark again the row and column will be '0'.

→ Rather go through entire column/row and mark them as '1'.

1	-1	-1	1
-1	0	0	-1
-1	-1	0	-1
1	-1	-1	1

→ Now do one more iteration and convert them into zero.

Pseudo code

```
for(i=0; i<n; i++)
{
    for(j=0; j<m; j++) ] n*m
    {
        if (arr[i][j] == 0)
        {
            markRow(i) → n
            markCol(j) → m
        }
    }
    markRow(i) → row no
}

for(j=0; j<m; j++) // Iterate for every col
{
    if (arr[i][j] != 0)
```

```

        arr[i][j] = -1;
    }
}
mark col (j)
{
    for(i=0; i <n ; i + e) // Iterate for over row
    {
        if (arr[i][j] != 0)
            arr[i][j] = -1;
    }
}

```

// Now iterate over the matrix if it's -1 \Rightarrow 0

```

for(i=0 → n)
{
    for(j=0 → m)
    {
        if (arr[i][j] == -1)
            arr[i][j] = 0;
    }
}

```

$$T.C \Rightarrow (n \times m) \times (n+m) + (n \times n) \quad S.C \Rightarrow O(1)$$

\approx power of cube.

2) Better Solution:

0	01	01	0
01	1	0	1
01	1	1	0

\rightarrow Col \rightarrow m.size

0/1	1	1	0	1
-2/1	1	0	0	1

↳ row → n.size

- if any row / col has minimum of one '0', mark the col / row.
- keep a track of '0'. So you can mark the row / col.
- So now there will be four size array for row / col.
- Initially the array value be '0'.
- Start iterating.
- Whenever '1' \Rightarrow ignore
- '0' \Rightarrow mark '1' in the array, for both row / col array.
- At the end we marked the row / col array as '1'.
- Now - do a reiteration whenever we are at $\boxed{1}$
 - will it be '0'? if only be '0' if the arr row / col is marked as '1'.
- If not marked '1' then the original array will be '1' only.
- Do the same for everything

0	1	1	0
0	1	10	10

0	1	10	10	1
1	1	10	10	10
1	10	10	10	10
1	1	10	10	10

$\{ \text{col}[m] = 10 \} \quad \text{row}[n] = 10 \}$

for ($i = 0 \rightarrow n$)

 for ($j = 0 \rightarrow m$)

 if ($\text{arr}[i][j] == 0$)

$\text{row}[i] = 1$ // both have been marked '1'
 $\text{col}[j] = 1$

 }

// Again reiterate. convert $1 \rightarrow 0$

for ($i = 0 \rightarrow n$)

 for ($j = 0 \rightarrow m$)

 if ($\text{row}[i] \text{ || } \text{col}[j]$)

$\text{arr}[i][j] = 0$

 }

CodeStudio Solution



✗

LeetCode Solution



✗



✗

T.C $\Rightarrow O(2^{kn} \cdot m)$
S.C $\Rightarrow O(n) + O(m)$

3rd Approach

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

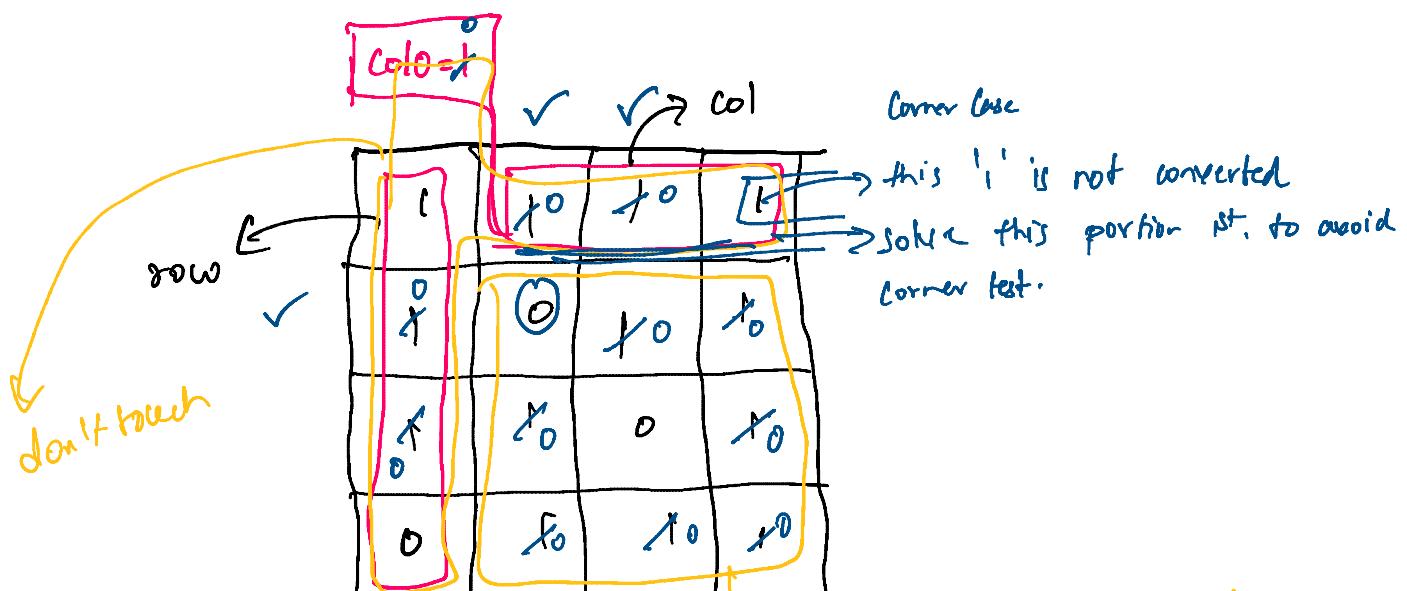
- keep a track in the matrix itself
- Consider 1st column to be the row

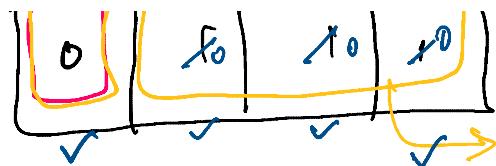
$$\text{col}[0] \rightarrow \text{row}[n]$$

- same 1st row to be the col

$$\text{row}[0] \rightarrow \text{col}[m]$$

But, it's NOT correct since matrix [0][0] is overlapping
 So, take separate variable col0, now there won't be
 any collision.





- Start iterating , once you see '0' keep a track and convert that row/col to zero
- Now after completing don't touch that matrix and

$T.C \Rightarrow O(n \times m)$ $S.C \Rightarrow O(1)$

