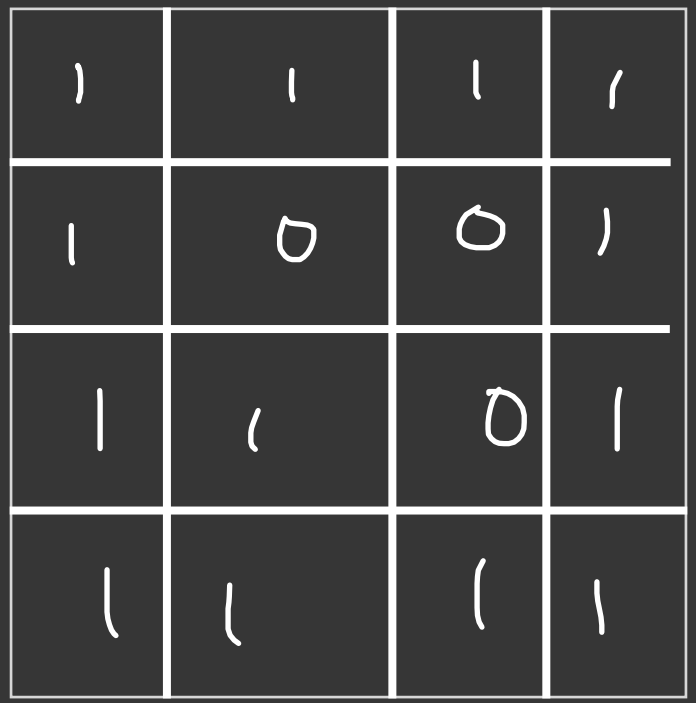
Set Matrix Zero:

Brute Force Approach:



→ Traverse for every zero we have mark or make it row and col zero

Brute force:

- We mark entire row and col when zero is with -1 in order to avoid zero conflict

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

→

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

- Then once again iterate and mark -1 with zero

Ans

T.C: $O(N*M) * O(N+M) + O(N*M) \rightarrow \approx O(N^3)$

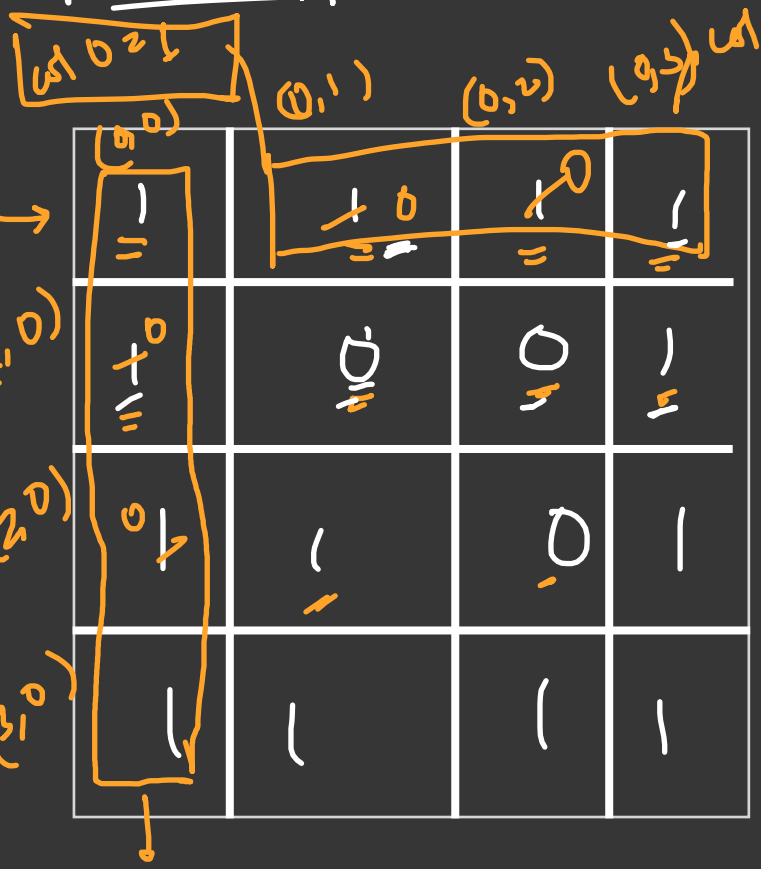
Better Approach:

- In this approach we know that a row/entire row or col would be zero if their exist minimum 1 Zero
- We reuse space row and col. arr. to keep track of zero
- We once traverse and the matrix and mark the row and col when we come across a Zero

T.C. $O(2 N*M)$

S.C. $O(N) + O(M)$

Optimal Approach:



In better approach we were using separate row and col. to mark the row and col at which our index of the matrix is zero

In optimal approach to optimize the space we use the matrix first row and col. as marking array

But here problem is for row[0] and col[0] we have same index so, which was not there in separate array

row = mat[i][0]
col = mat[0][j] = 0

if(mat[i][j] = = 0)
mark row = mat[i][0] = 0
mark col = mat[0][j] = 0

After marking

We iterate matrix and avoiding 1 row and col to avoid wrongly setting zero in matrix

```
for(i=1, i<n; i++)
    for(j=1, j<m; j++)
```