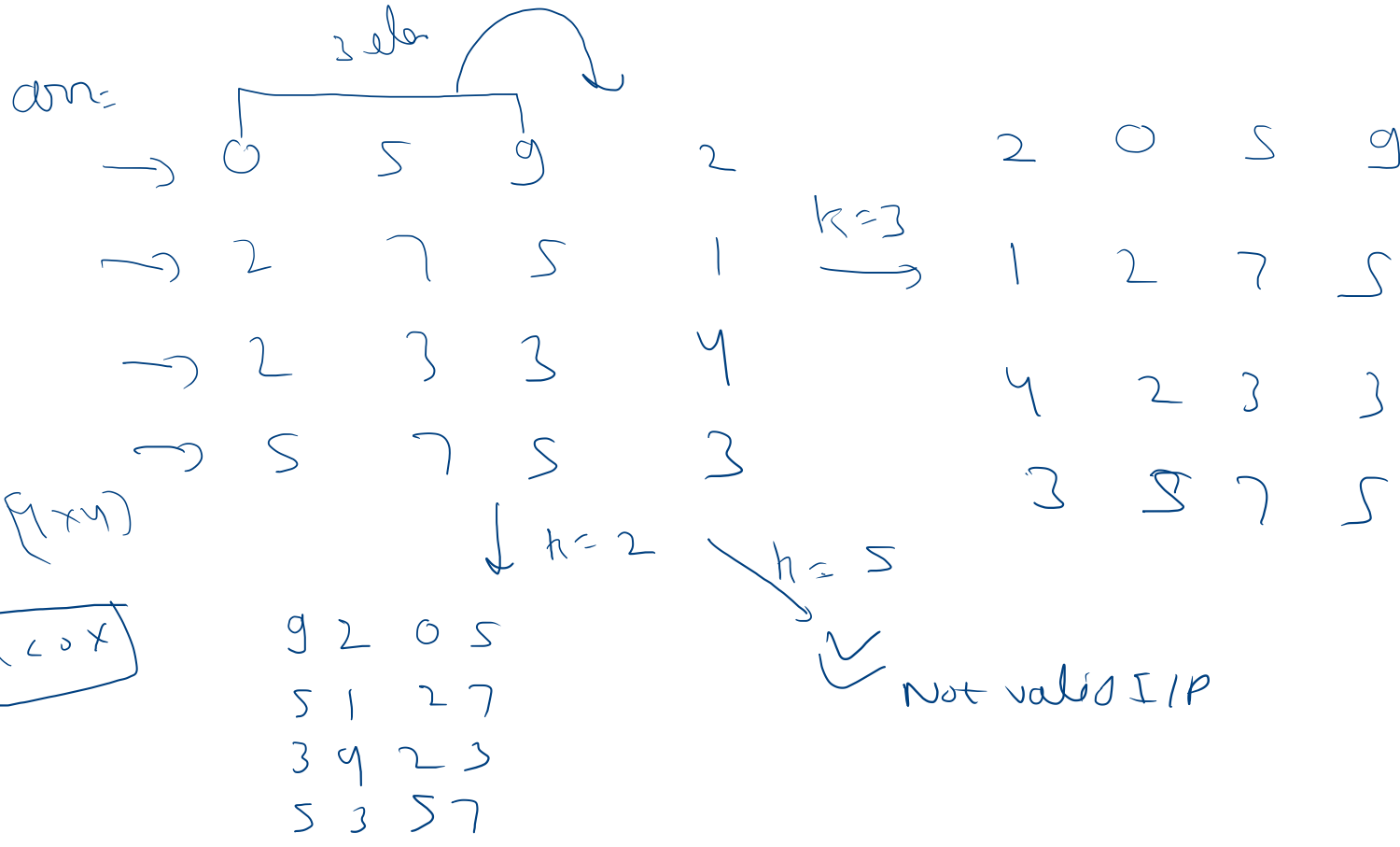
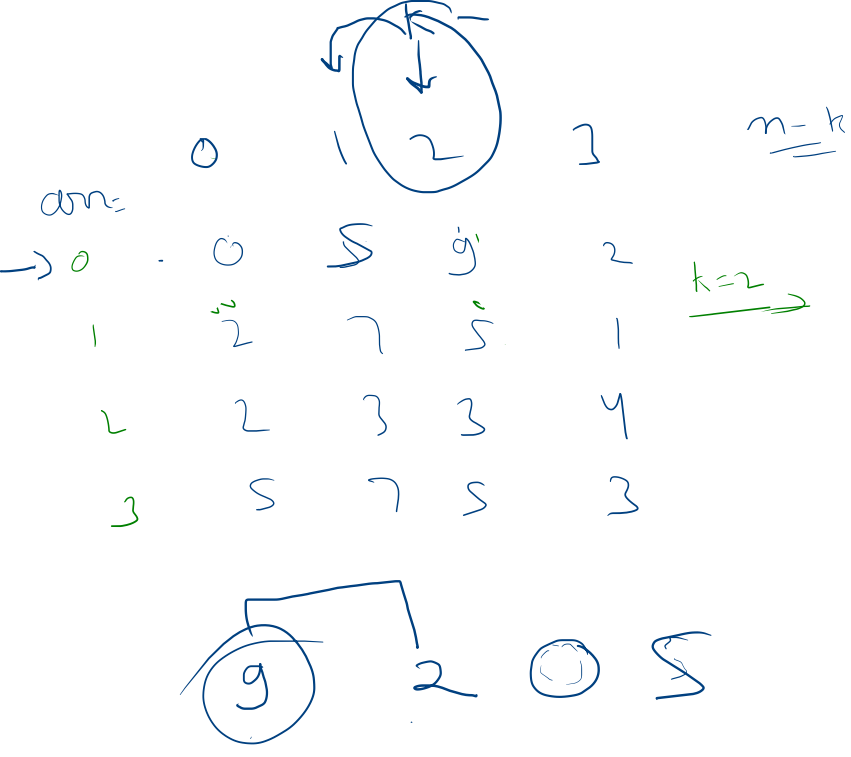
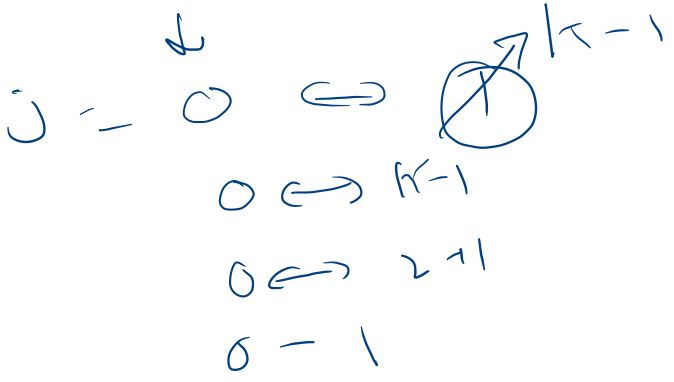
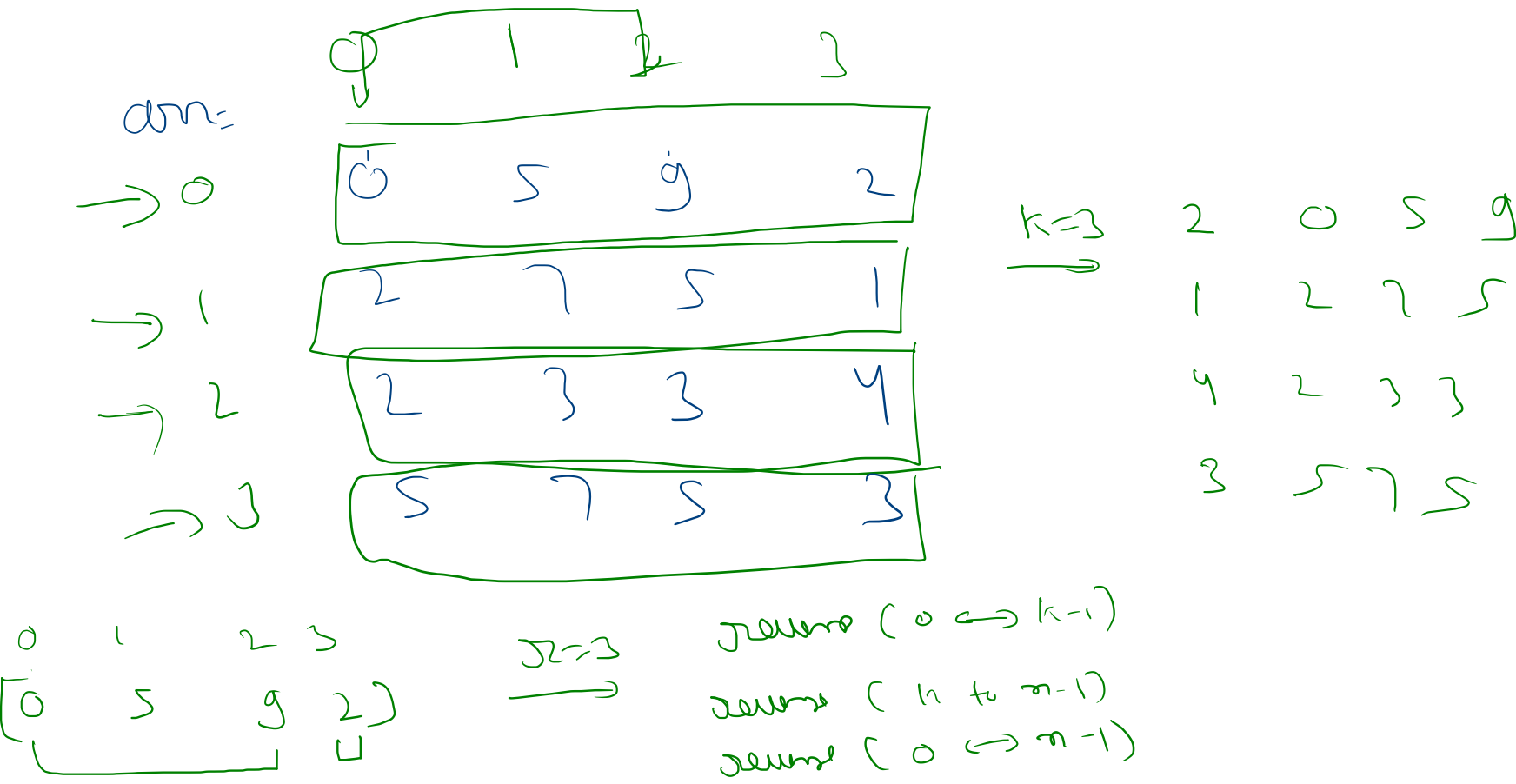
Shift Matrix Row-Wise (21 july)

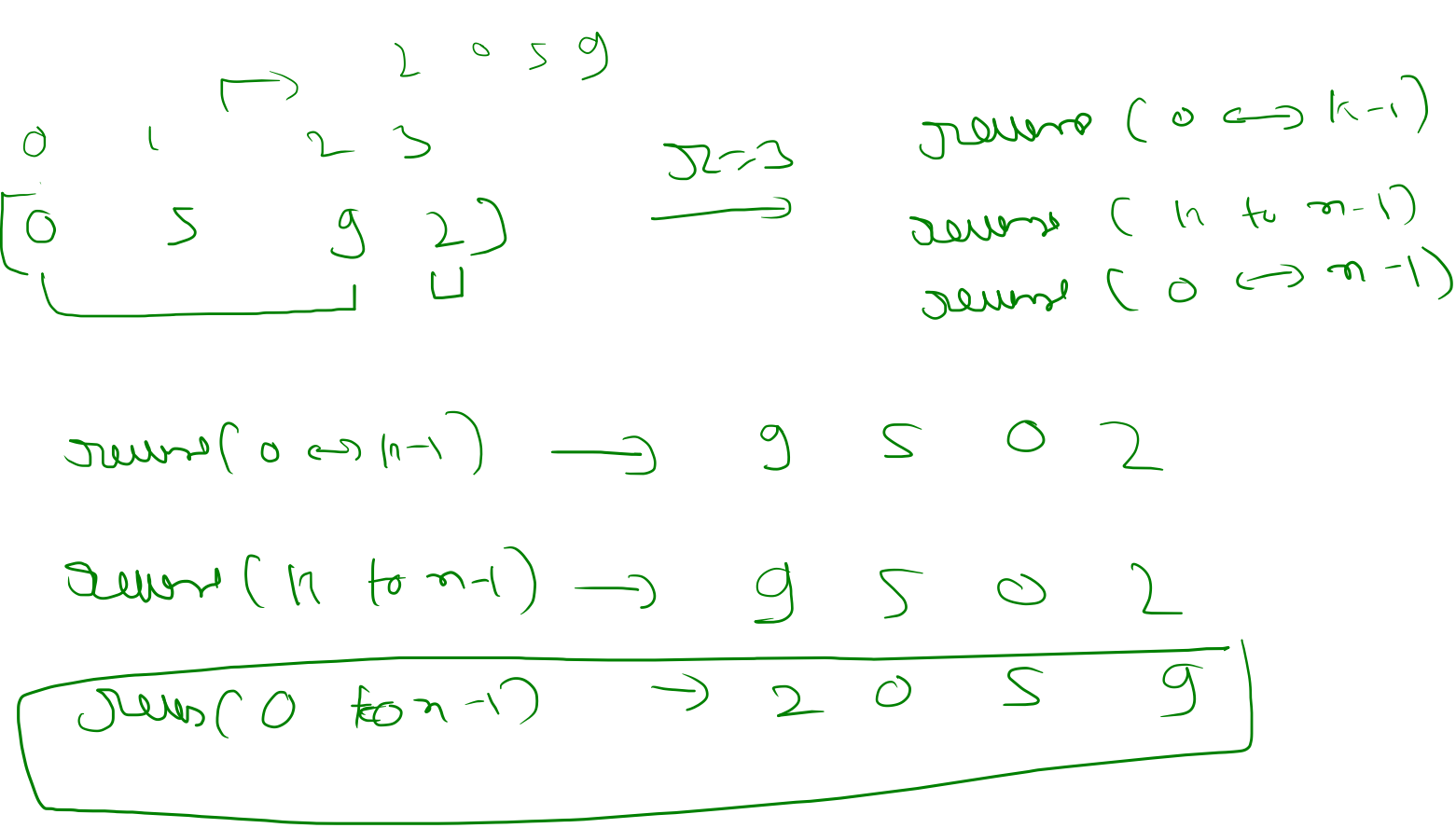




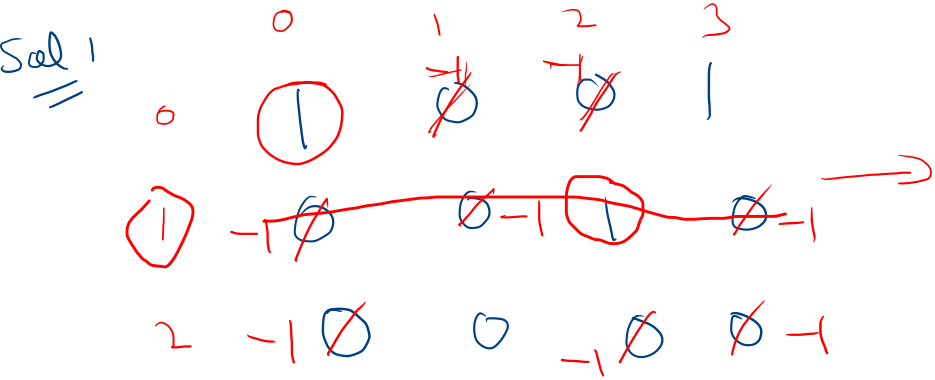
```
for (int row = 0; row < arr.length; row++) {

    // print
    for (int j = k; j < arr[row].length; j++) {
        System.out.print(arr[row][j] + " ");
    }

    for (int j = 0; j < k; j++) {
        System.out.print(arr[row][j] + " ");
    }
    System.out.println();
}
```



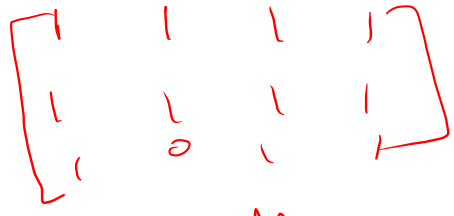

0 1 2 3
arr:
0 0 5 9 2
1 2 7 5 1
2 2 3 3 4
3 5 7 5 3
k=3
2 0 5 9
1 2 7 5
4 2 3 3
3 5 7 5
0 1 2 3
0 5 9 2
k=3
recurse (0 ↔ k-1)
recurse (k to n-1)
recurse (0 ↔ n-1)


2 0 5 9
0 1 2 3
[0 5 9 2]
i=3
reverse (0 ↔ k-1)
reverse (k to n-1)
reverse (0 ↔ n-1)
reverse(0 ↔ n-1) → 9 5 0 2
reverse(k to n-1) → 9 5 0 2
reverse(0 to n-1) → 2 0 5 9

Modify The Matrix (21 july)



Traverse



S.1 Traverse all elements

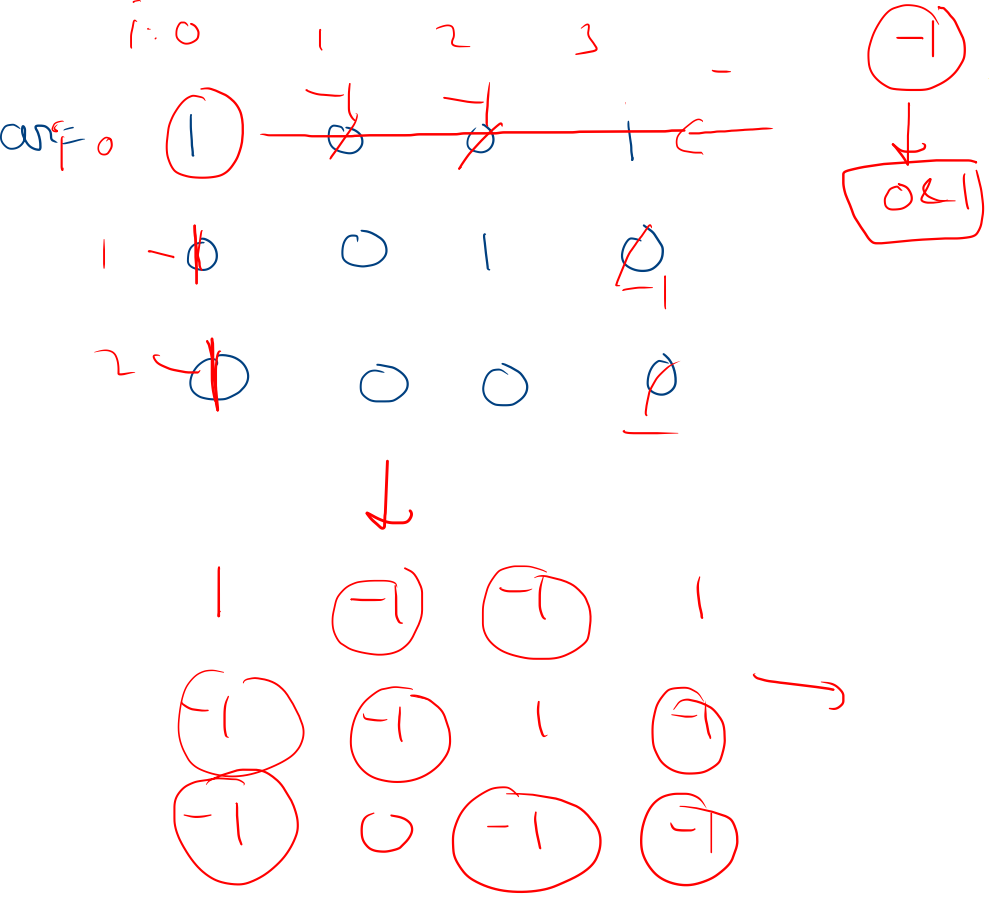
S.2 if (arr[i][j] == -1) {

i → -1

j → -1

}

diff (0,1) → Replace all non 0 with -1



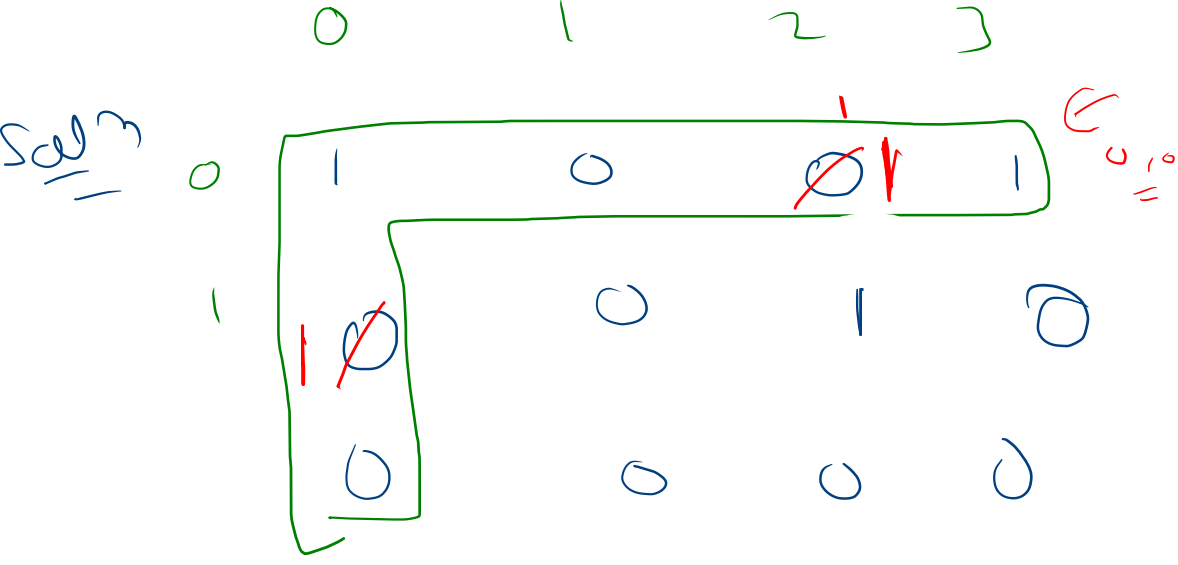
```java
for (int i = 0; i < arr.length; i++) {

    for (int j = 0 ; j < arr[i].length; j++) {

        if (arr[i][j] == 1) {
            replaceWithOne(arr, i, j);
        }

    }
```

```java
public static void replaceWithOne(int[][] arr, int i, int j) {

    // Column
    for (int row = 0; row < arr.length; row++) {
        if (arr[row][j] != 1) {
            arr[row][j] = -1;
        }
    }

    // row
    for (int col = 0; col < arr[0].length; col++) {
        if (arr[i][col] != 1) {
            arr[i][col] = -1;
        }
    }
```

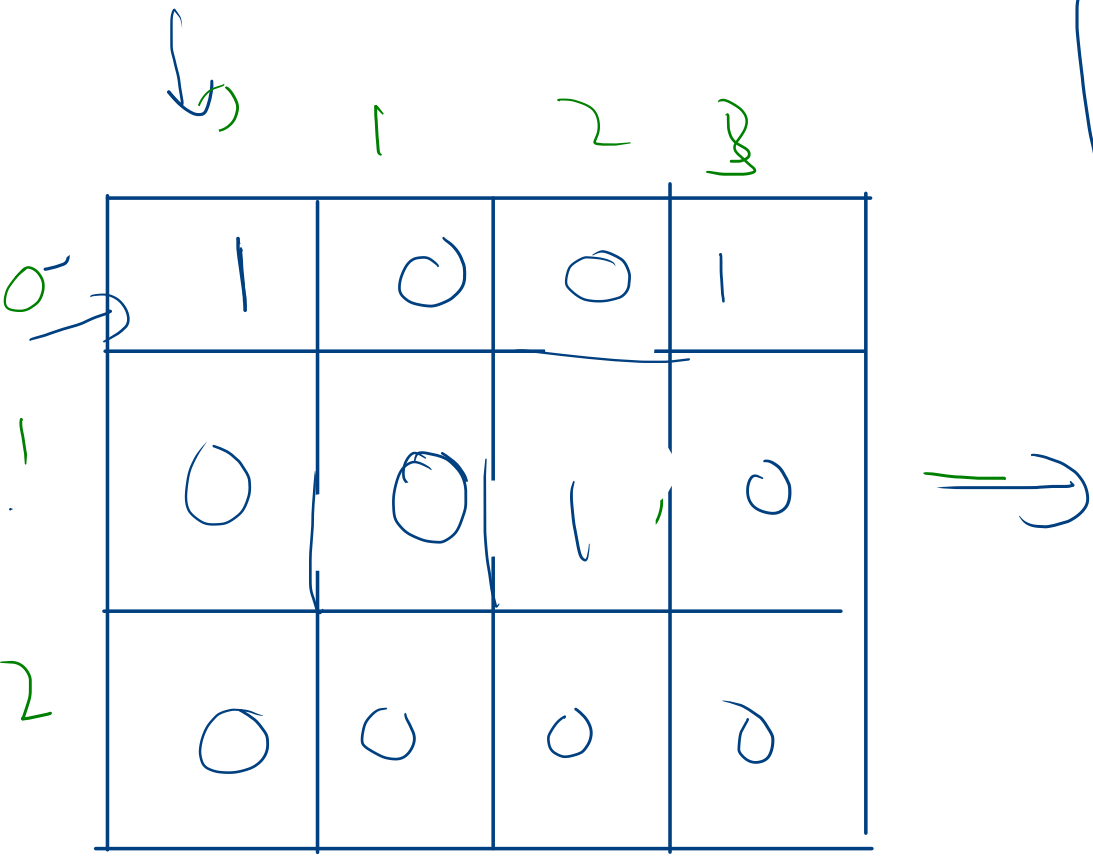
Soln 1

Time comp $\Rightarrow O(m \times n(m+n))$

space $\Rightarrow O(1)$

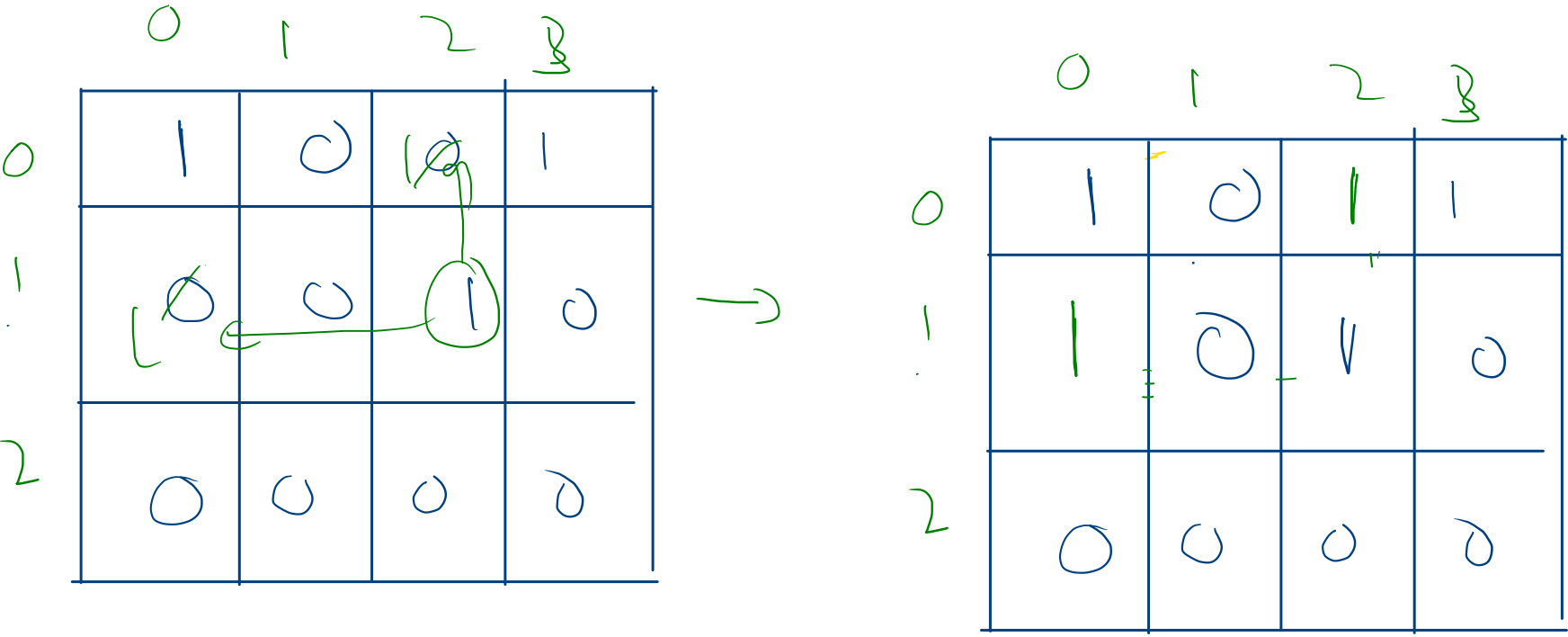


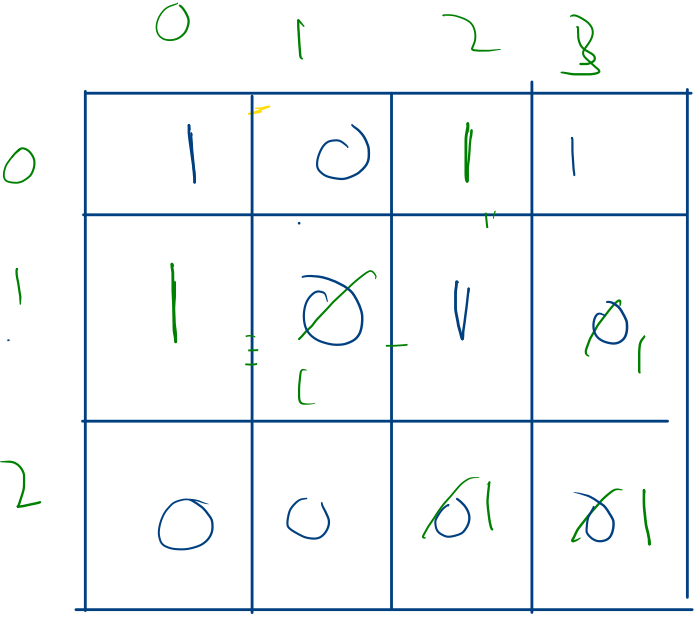
- Find row & first col elem for each ckt
  or
  last row & last col elem for ---

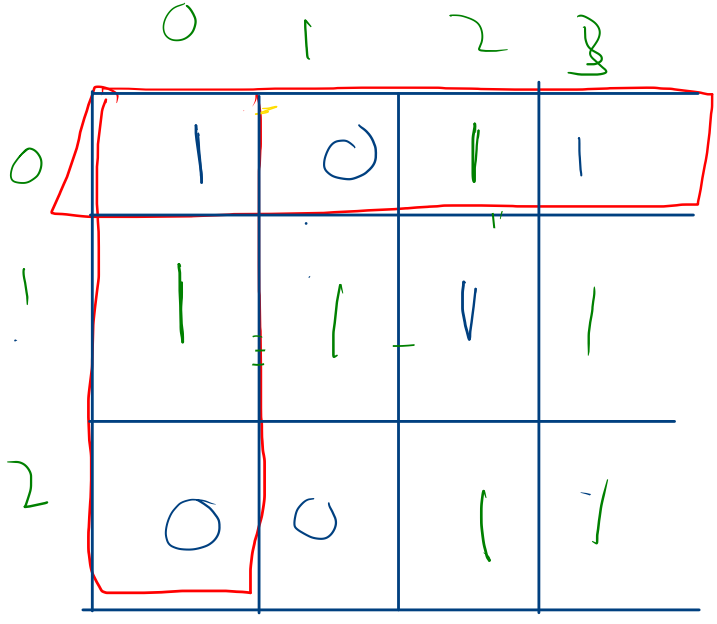


isRow = true

isCol = true







isRow == true → mark all the → 1

isCol == true → mark ith col the = 1