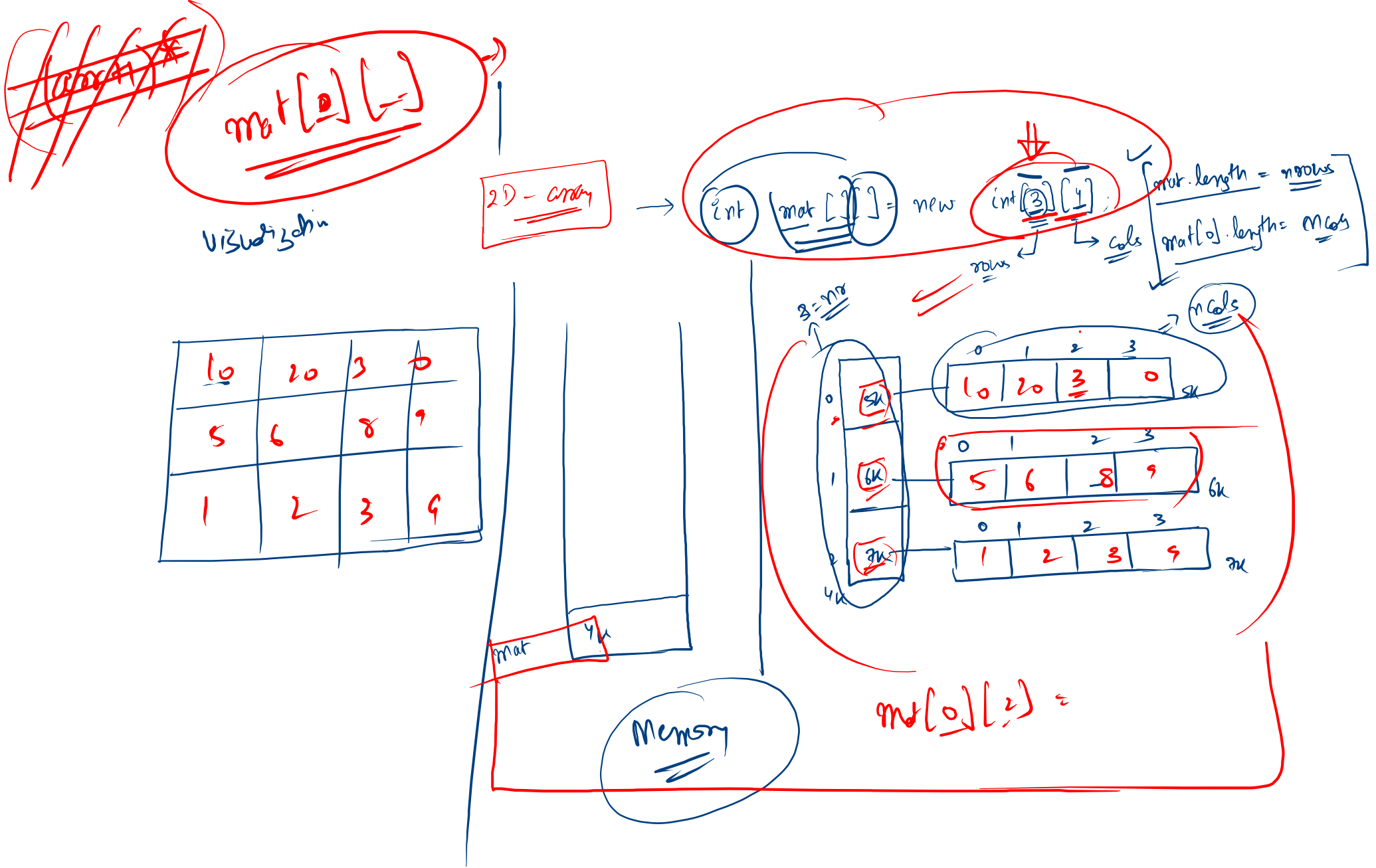
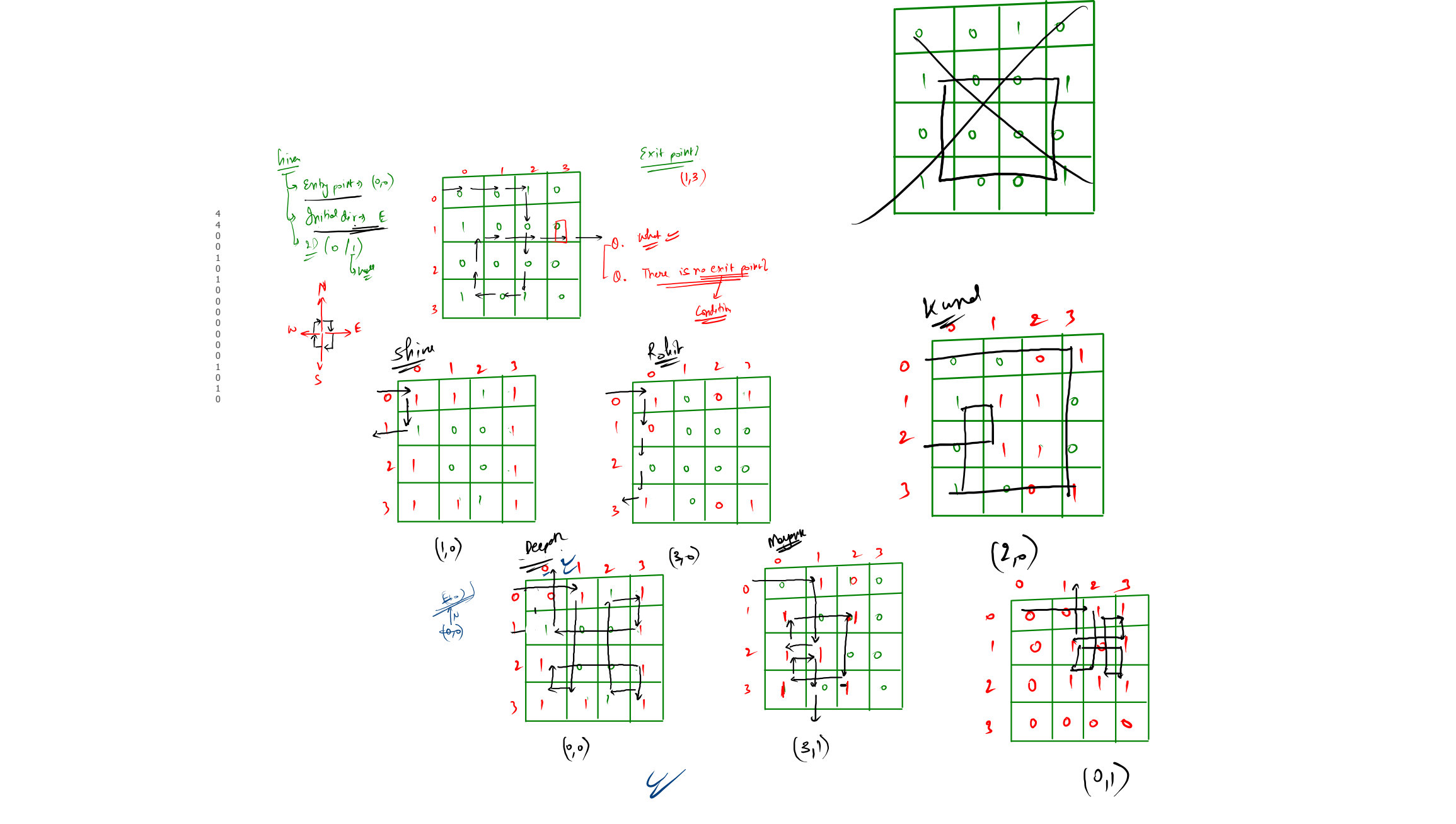
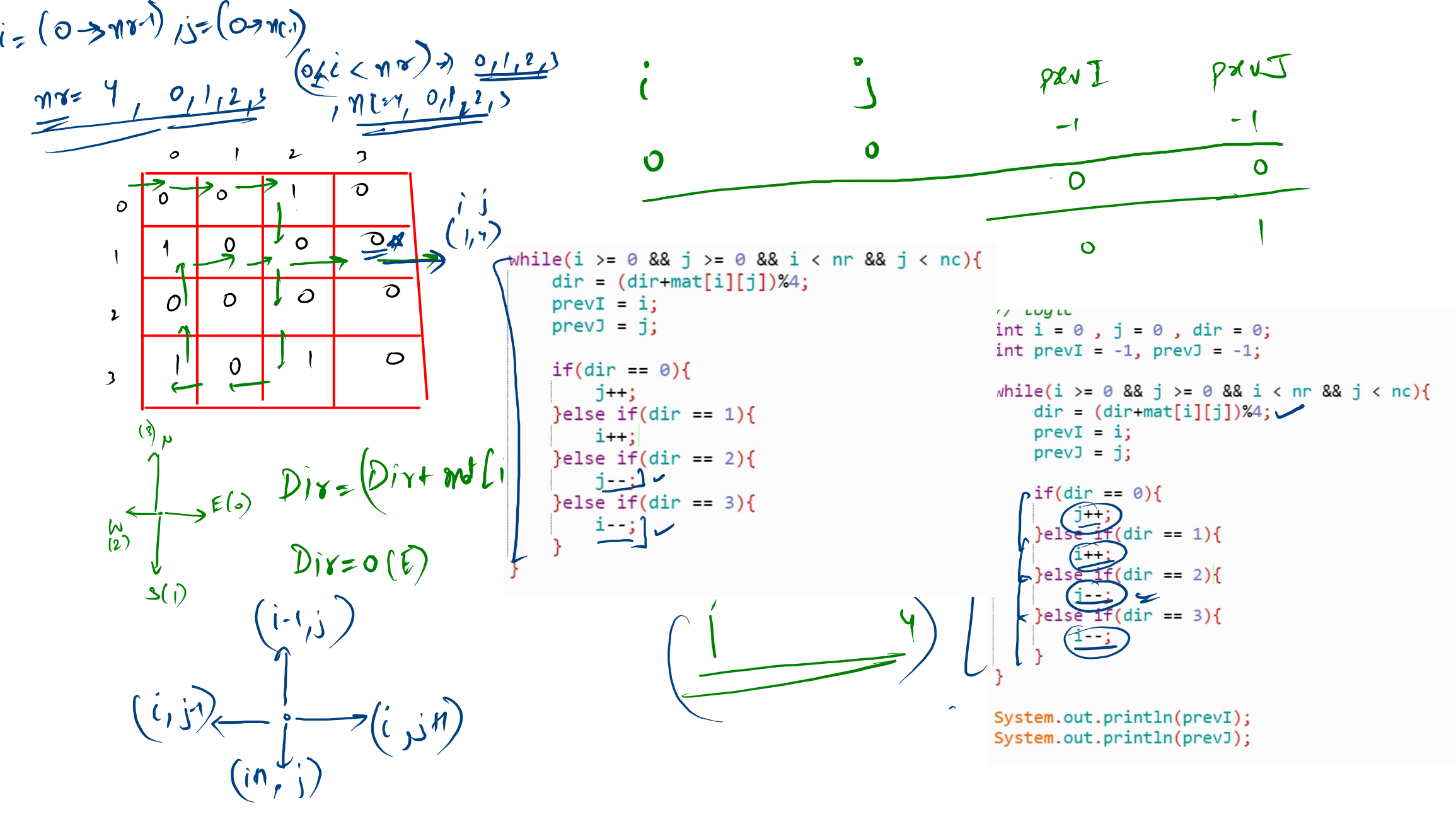
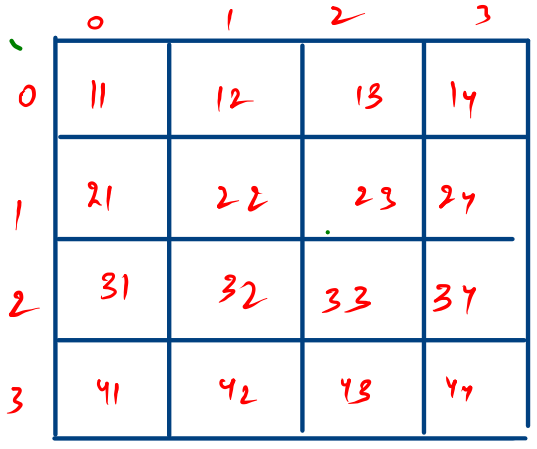

mat[0][2]
Visualization
2D - array
int mat[][] = new int[3][4];
rows
cols
mat.length = nRows
mat[0].length = nCols
10 20 3 0
5 6 8 9
1 2 3 9
mat
4K
Memory
nCols
nRows
0 1 2 3
10 20 3 0
5 6 8 9
1 2 3 9
mat[0][2] =


Shiva
Entry point (0,0)
Initial dir E
2D (0/1)
Exit point
(1,3)
What
There is no exit point
Condition
N
W
E
S
Shiva
(1,0)
Rohit
(3,0)
Kunal
(2,0)
Deepak
(0,0)
Mayank
(3,1)
(0,1)


i = (0 -> nr-1) j=(0->nc-1)
nr= 4, 0,1,2,3
(0<=i<nr) -> 0,1,2,3
nc=4, 0,1,2,3
i
j
prevI
prevJ
-1
-1
0
0
0
0
0
1
0
0 1 2 3
0
1
2
3
(1,4)
while(i >= 0 && j >= 0 && i < nr && j < nc){
dir = (dir+mat[i][j])%4;
prevI = i;
prevJ = j;
if(dir == 0){
j++;
}else if(dir == 1){
i++;
}else if(dir == 2){
j--;
}else if(dir == 3){
i--;
}
}
int i = 0 , j = 0 , dir = 0;
int prevI = -1, prevJ = -1;
while(i >= 0 && j >= 0 && i < nr && j < nc){
dir = (dir+mat[i][j])%4;
prevI = i;
prevJ = j;
if(dir == 0){
j++;
}else if(dir == 1){
i++;
}else if(dir == 2){
j--;
}else if(dir == 3){
i--;
}
}
System.out.println(prevI);
System.out.println(prevJ);
N(3)
W(2)
E(0)
S(1)
Dir=(Dir+mat[
Dir=0(E)
(i-1,j)
(i,j-1)
(i,j+1)
(i+1,j)



No extra space

41 31 21 11
42 32 22 12
43 33 23 13
44 34 24 14

Square matrix → rows == col

4
11
12
13
14
21
22
23
24
31
32
33
34
41
42
43
44

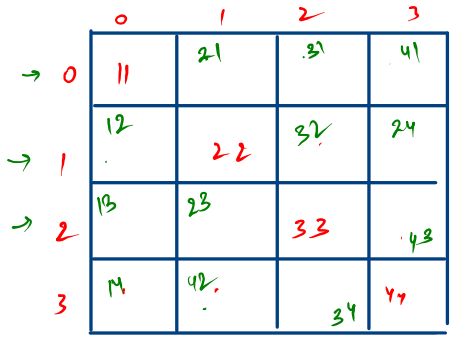
① Transpose

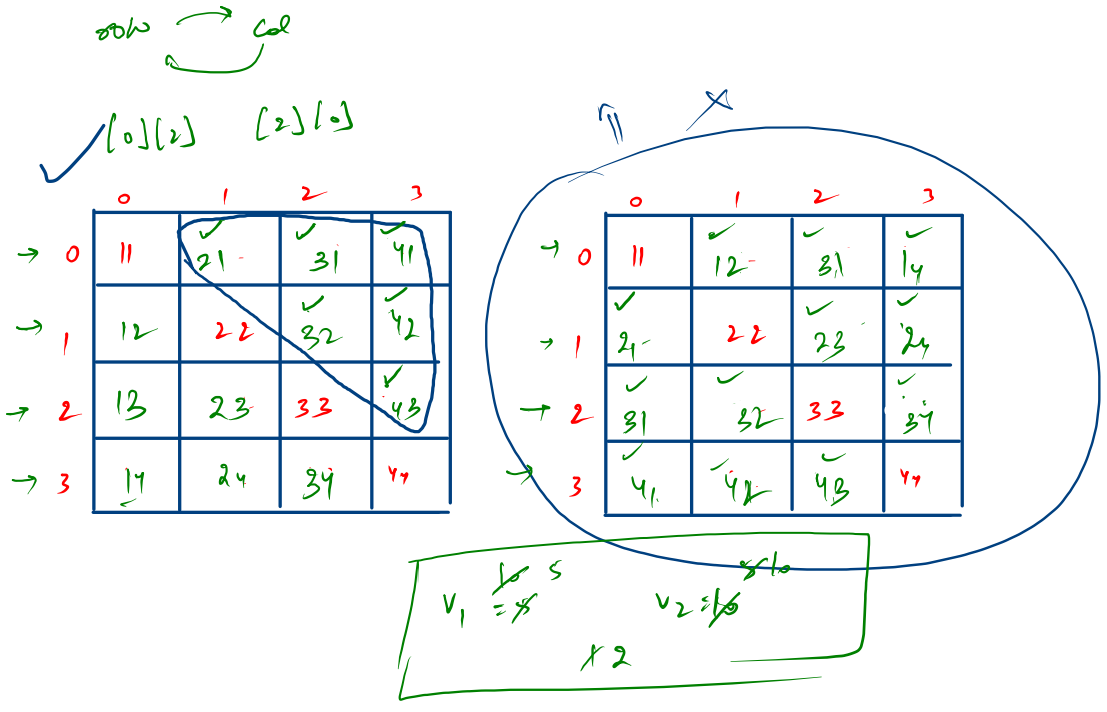
rows ↔ col

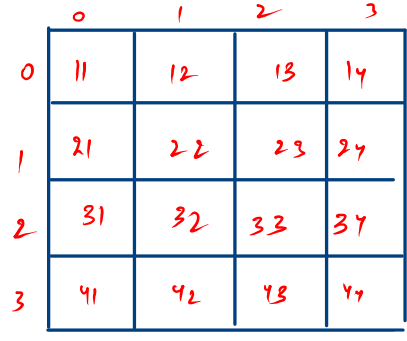
11	21	31	41
12	22	32	42
13	23	33	43
14	24	34	44

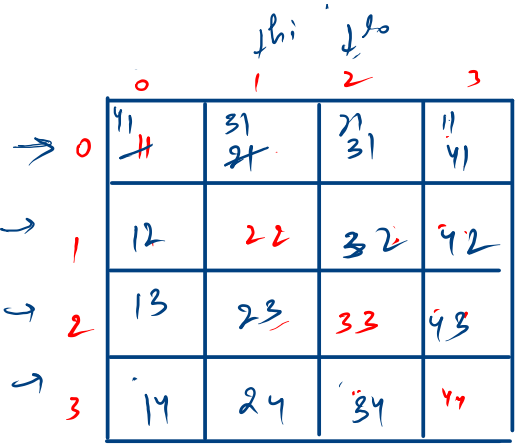
② reverse each row

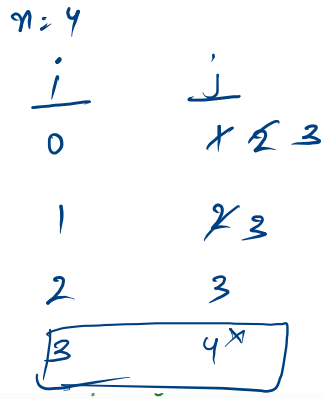
41	31	21	11
42	32	22	12
43	33	23	13
44	34	24	14


0 1 2 3
0 11 21 31 41
1 12 22 32 24
2 13 23 33 43
3 14 42 34 44


row
col
[0][2] [2][0]
0 1 2 3
0 11 21 31 41
1 12 22 32 42
2 13 23 33 43
3 14 24 34 44
0 11 12 31 14
1 21 22 23 24
2 31 32 33 34
3 41 42 43 44
V1 :
V2 :
x2

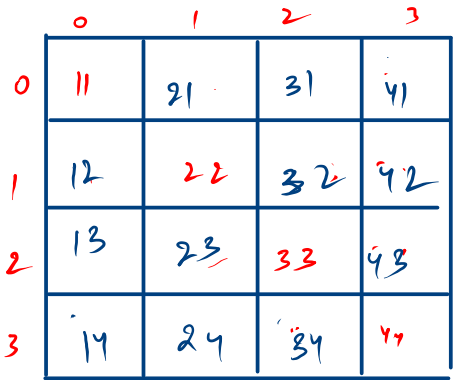

0 1 2 3
0 11 12 13 14
1 21 22 23 24
2 31 32 33 34
3 41 42 43 44



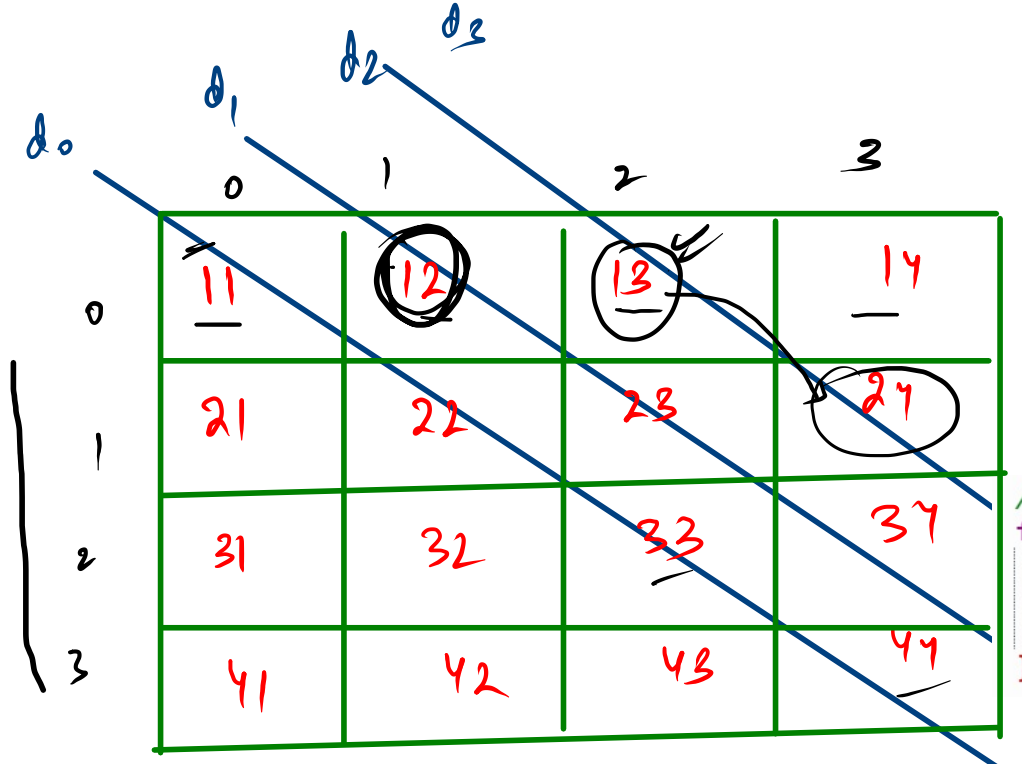


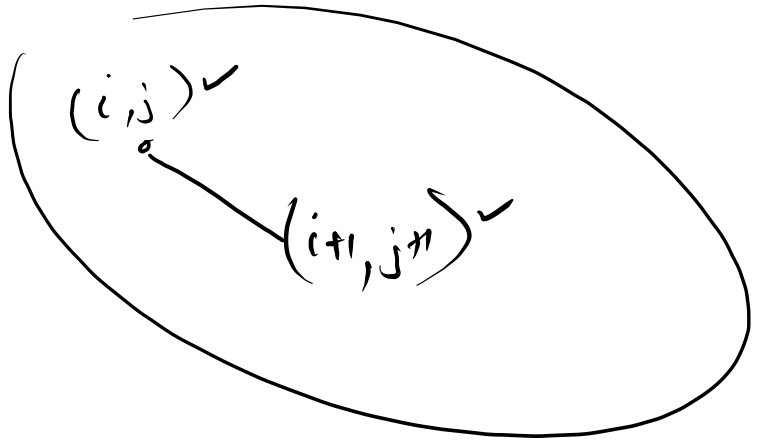

```java
int n = mat.length;

for(int i = 0 ; i < n ; i++){
    for(int j = i+1 ; j < n ; j++){
        int tmp = mat[i][j];
        mat[i][j] = mat[j][i];
        mat[j][i] = tmp;
    }
}
```



4 → nr, nc






```java
// Logic
for(int gap = 0  ; gap < n ; gap++){
    for(int r = 0 , c = gap ; r < n && c < n ; r++ , c++){
        System.out.println(mat[r][c]);
    }
}
```

n=4

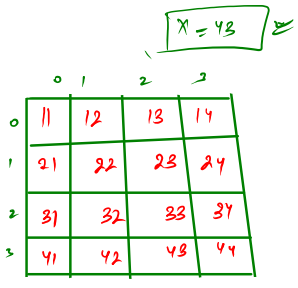
d0 → [0][0] [1][1] [2][2] [3][3] gap = 0

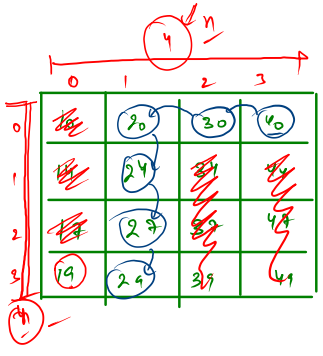
d1 → [0][1] [1][2] [2][3] gap = 1

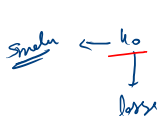
d2 → [0][2] [1][3] gap = 2

d3 → [0][3] gap = 3

gap	r	c	
0	0	0	→ 11
	1	1	→ 22
	2	2	→ 33
	3	3	→ 44
1	0	1	








```java
public static void main(String[] args) throws Exception {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();

    int mat[][] = new int[n][n];

    for(int i = 0 ; i < n ; i++){
        for(int j = 0 ; j < n ; j++){
            mat[i][j] = scn.nextInt();
        }
    }
    int x = scn.nextInt();
    // Logic
    int i = 0 , j = n-1;

    while(j >= 0 && i < n){
        if(mat[i][j] == x){
            System.out.println(i);
            System.out.println(j);
            return;
        }else if(x < mat[i][j]){
            j--;
        }else if(x > mat[i][j]){
            i++;
        }
    }
    System.out.println("Not Found");
}
```