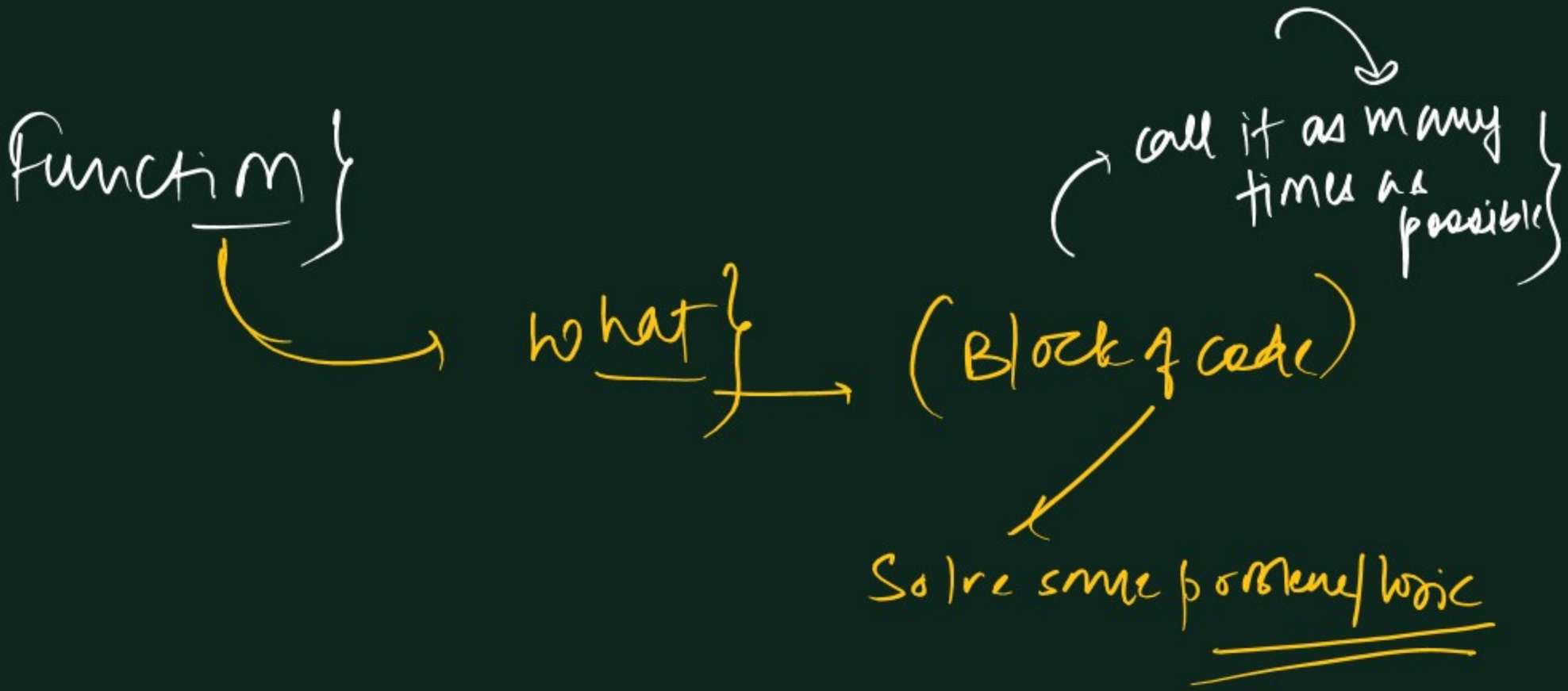
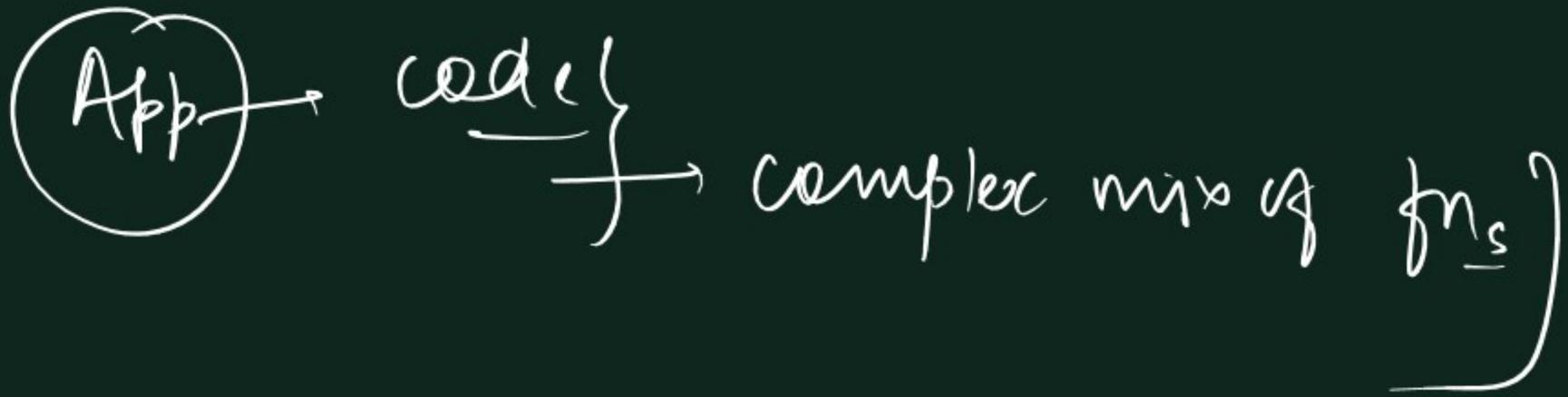

function
what
(Block of code)
call it as many times as possible
Solve some problem/logic

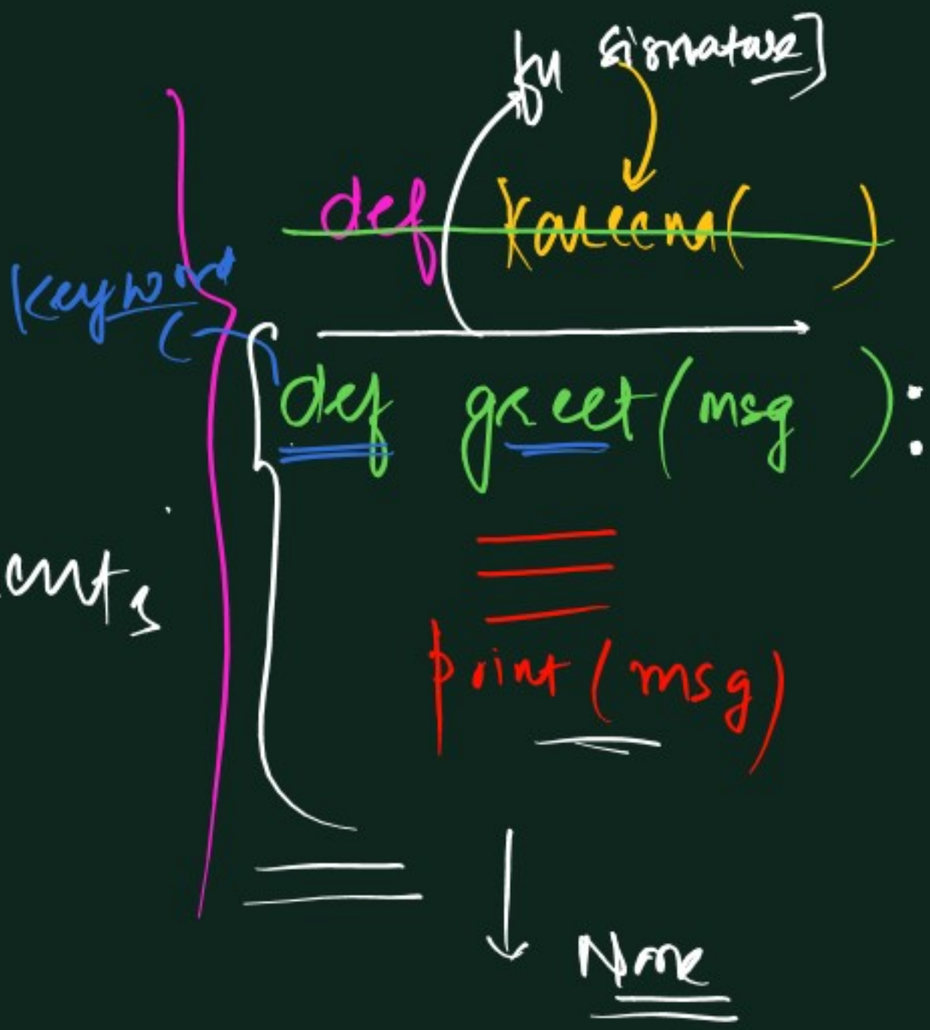

App
code
complex mix of fns

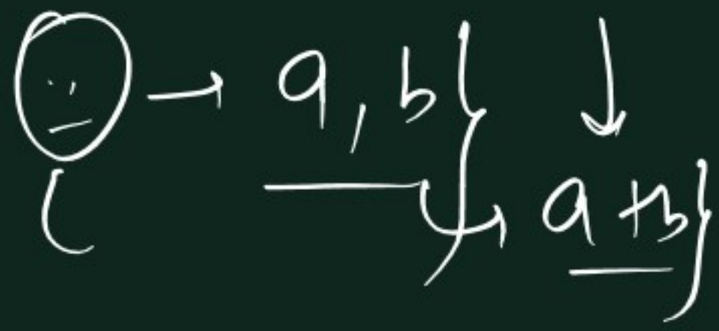
(fn)

① body → logic

② Parameter / Arguments

③ Return section



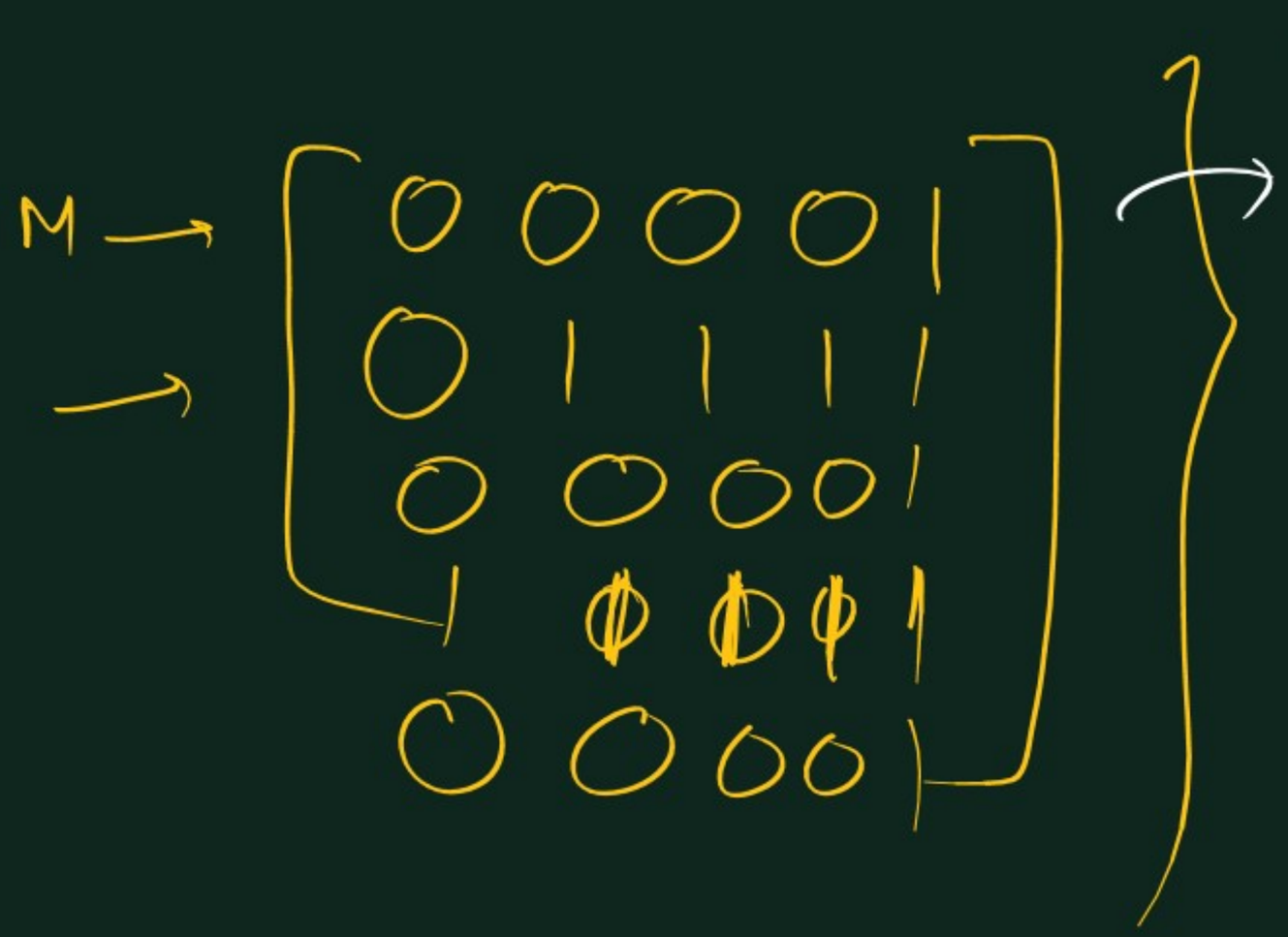


```
def sum(a, b):
    result = a+b
    return result
```

Return statement

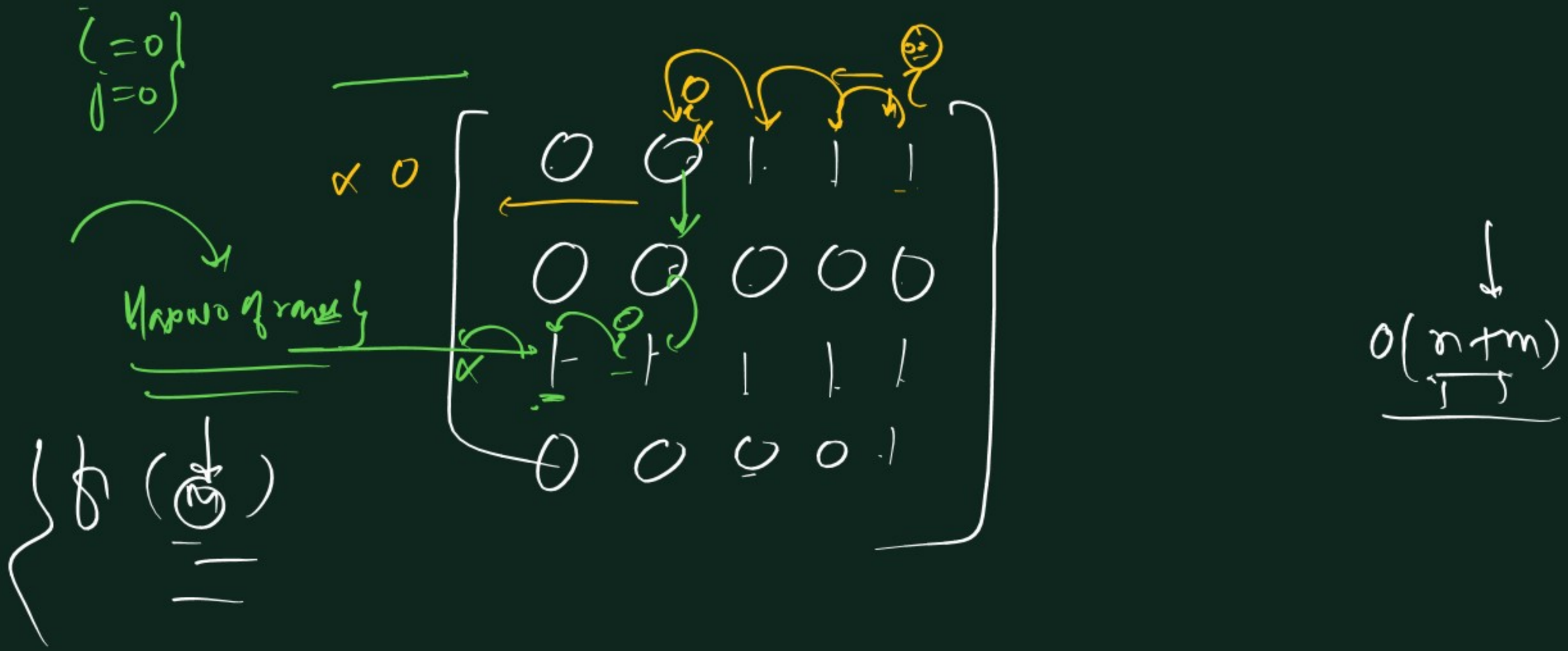
parameters

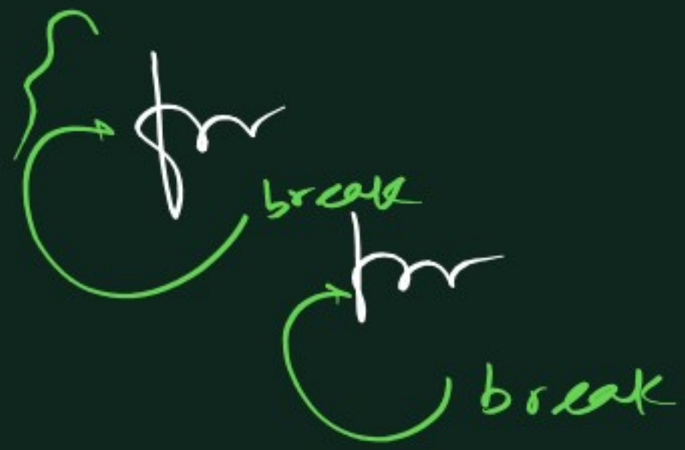
```
x = sum(5, 6)
print(sum(5, 6))
```

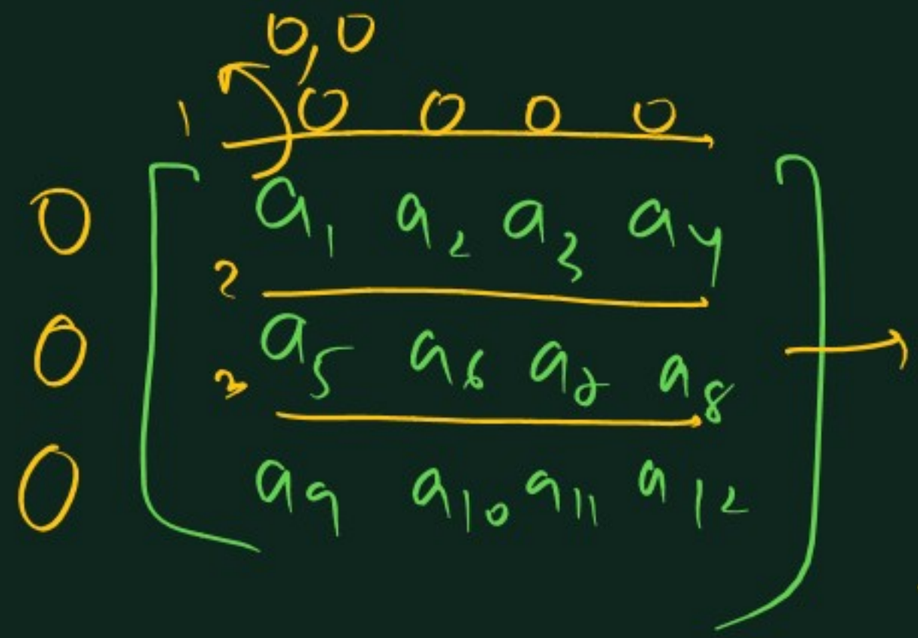



```
def findRowWithMaxOnes(matrix):
    ===
    ===
    ===

    return rowIndex
```

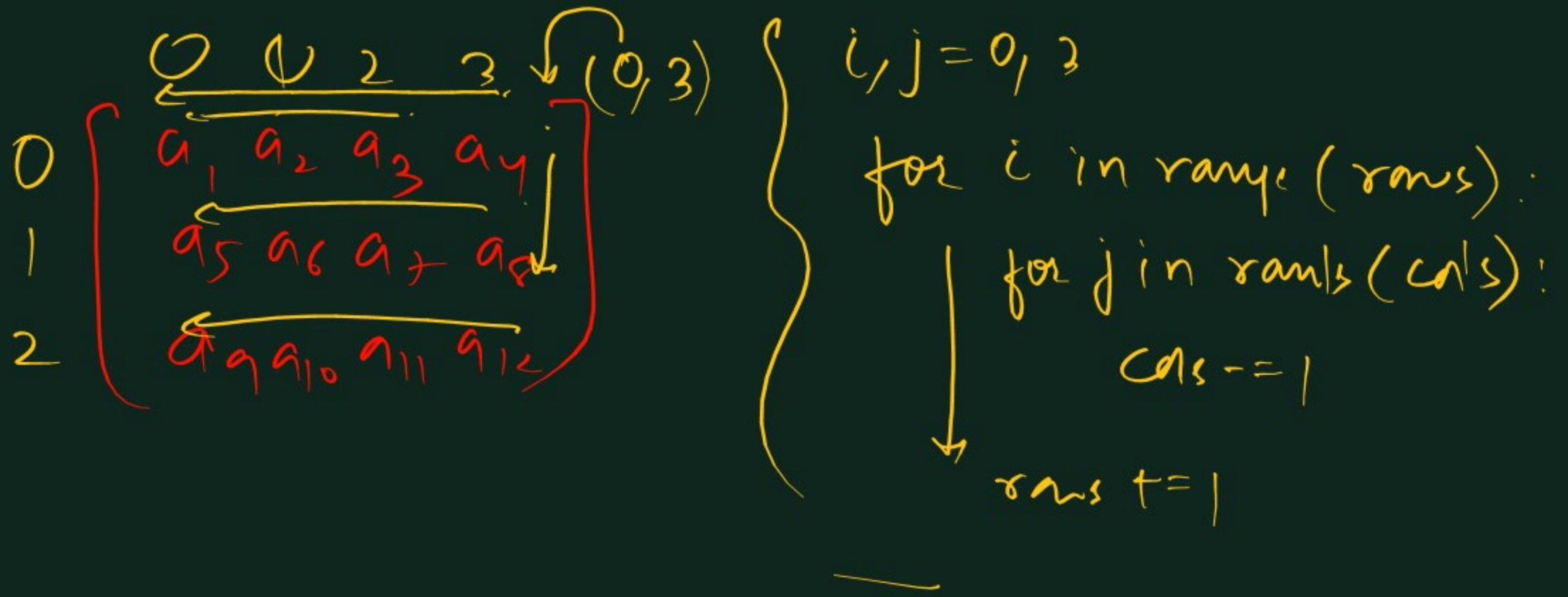

i=0
j=0
Mapping of rows
O(n+m)

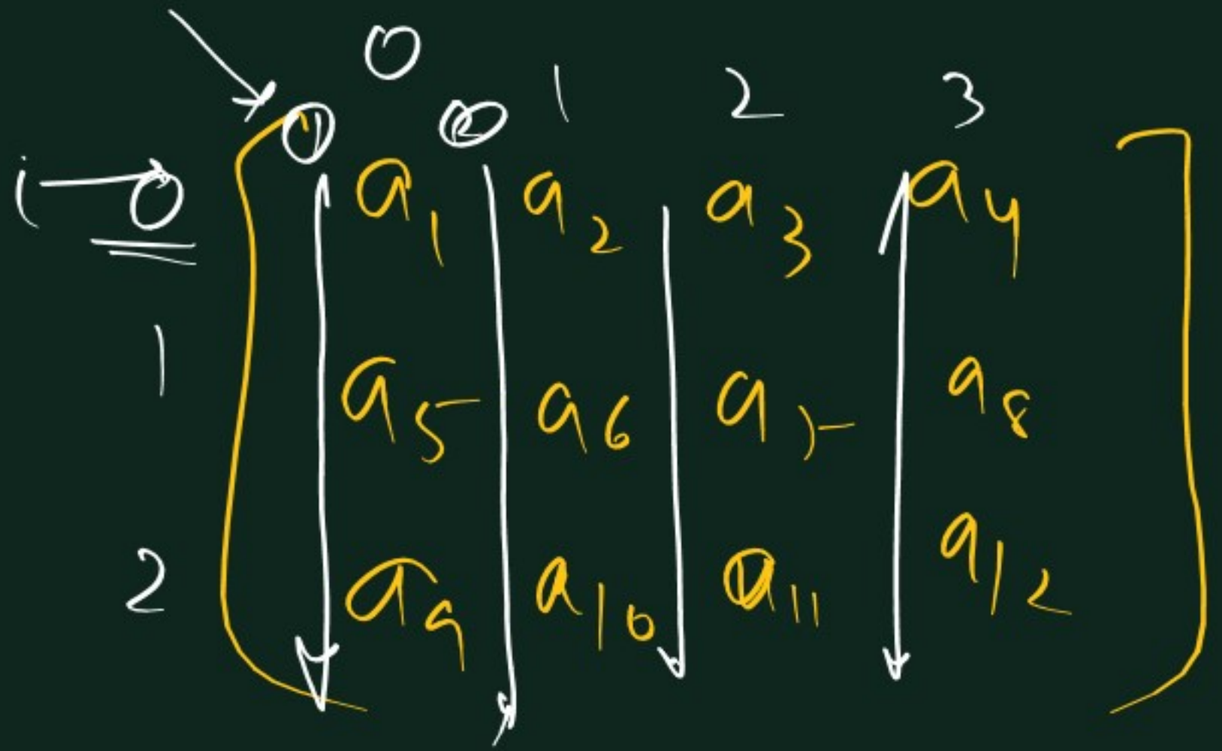

for
break
for
break




```
i, j = 0, 0
for i in range(rows):
    for j in rany(cols):
```

```
while (i < rows):
    while (j < cols):
```

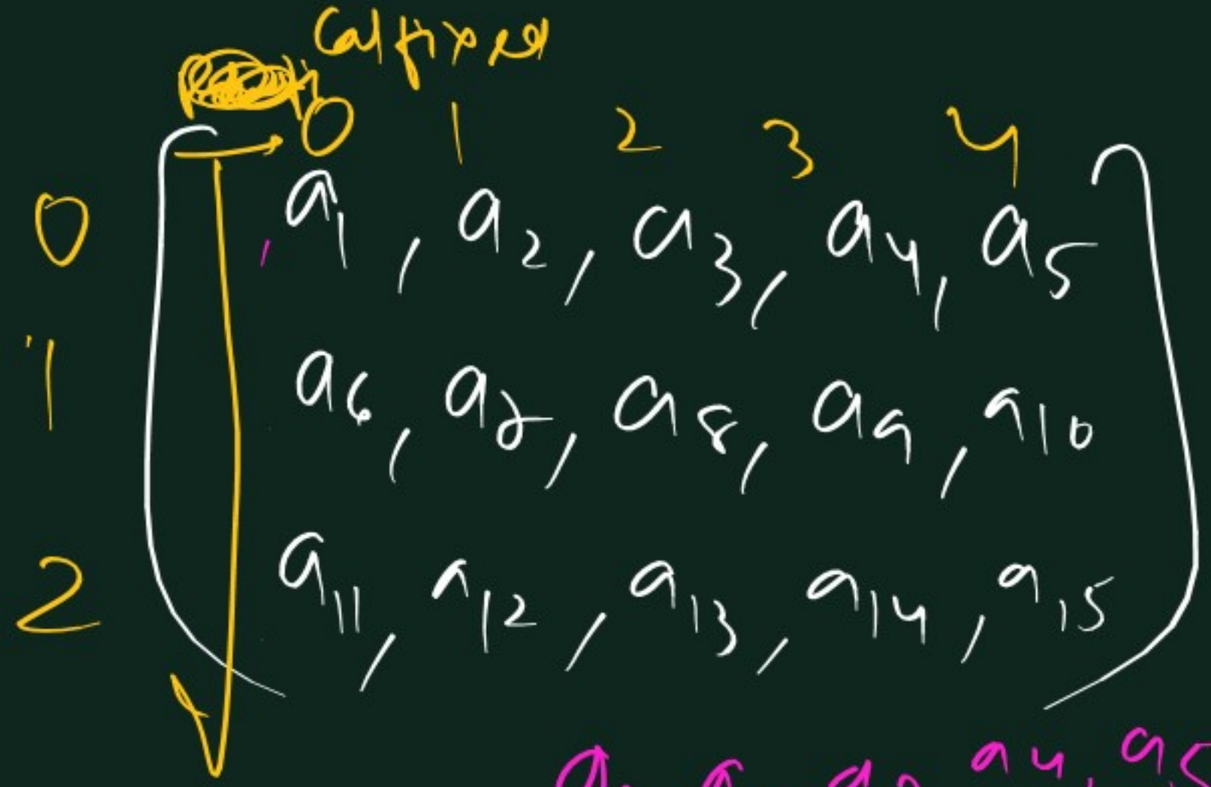

0 1 2 3 (0, 3)
0
1
2
a1 a2 a3 a4
a5 a6 a7 a8
a9 a10 a11 a12
i, j = 0, 3
for i in range(rows):
for j in range(cols):
cols -= 1
rows += 1



$a_1, a_5, a_9 \ldots$

$i, j = 0, 0$

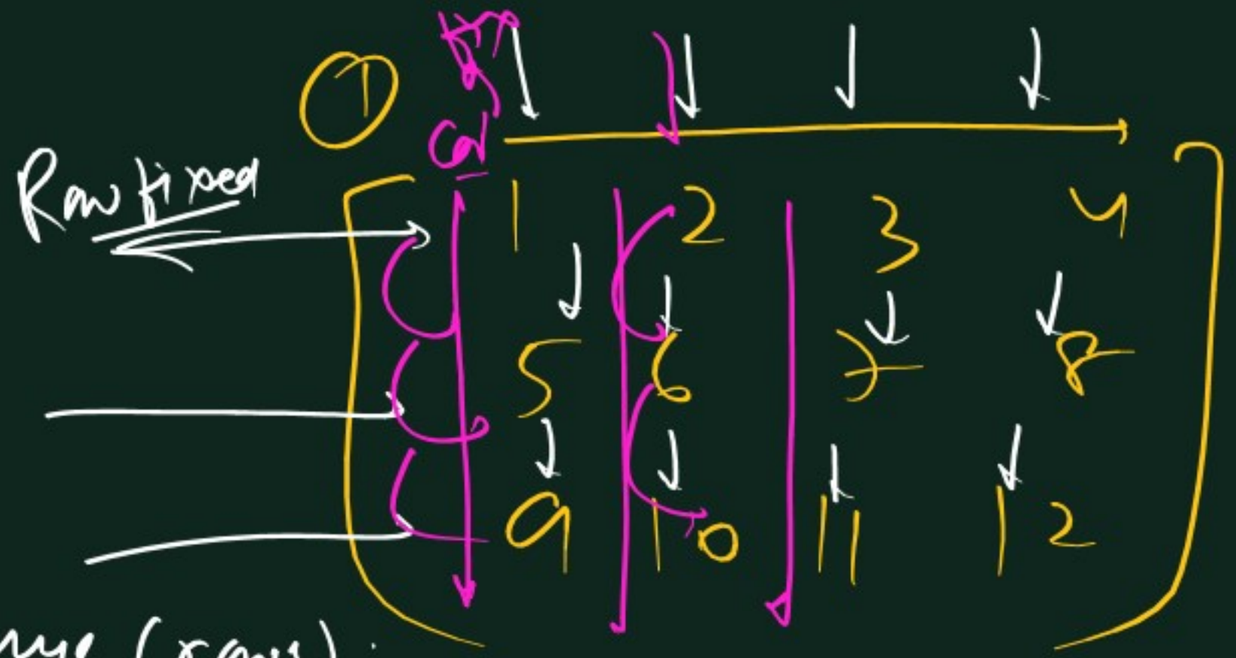
```
for j in range(cols):
    for i in range(rows):
        print(matrix[i][j])
    i += 1
j += 1
```



$a_1\ a_2\ a_3\ a_4\ a_5\ a_6\ a_7$

print(m[i][j])

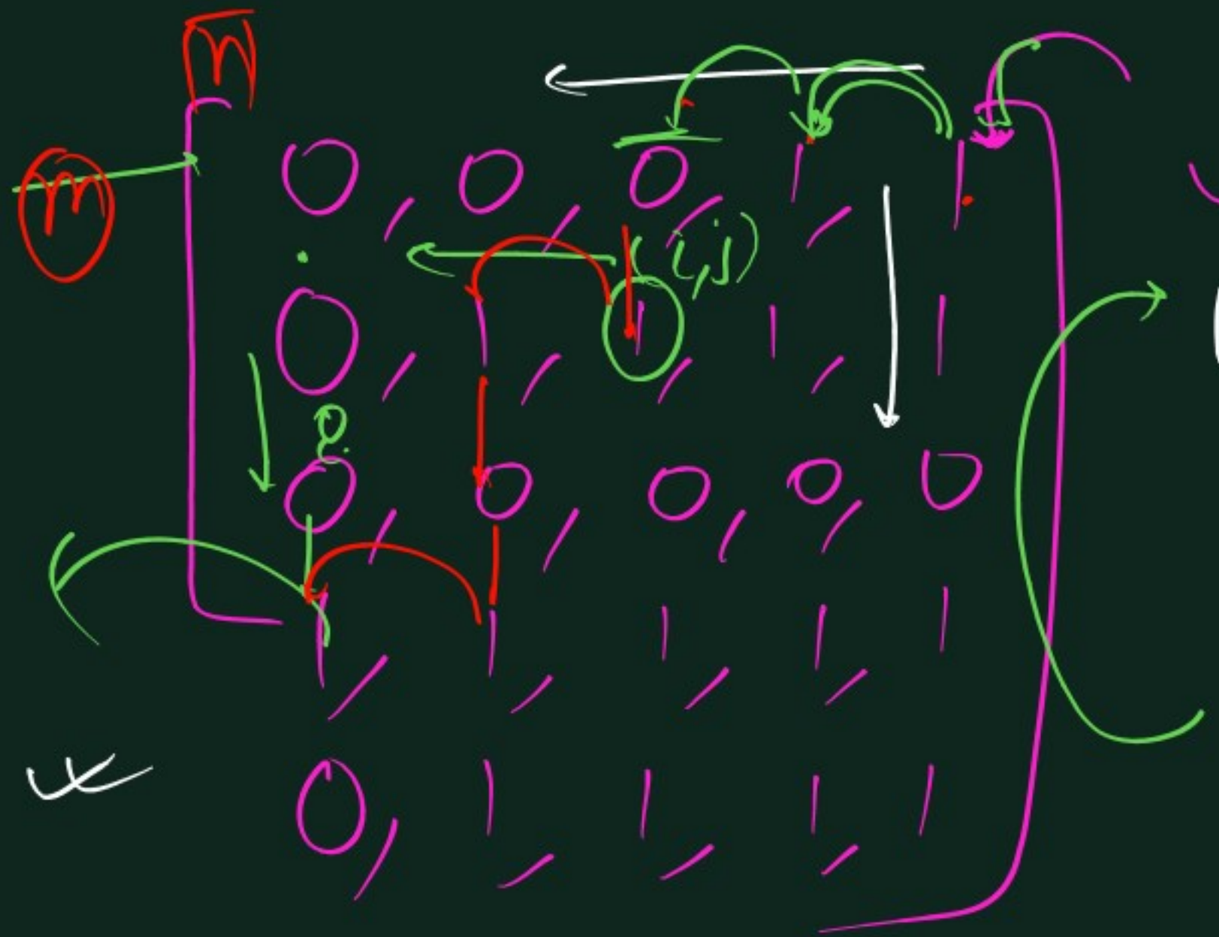
B-1



1 2 3 4 5 6, ...

```
for i in range(rows):
    for j in range(cols):
```

```
for ① in range(cols):
    [for j in range(rows)
        M[j][i]
```



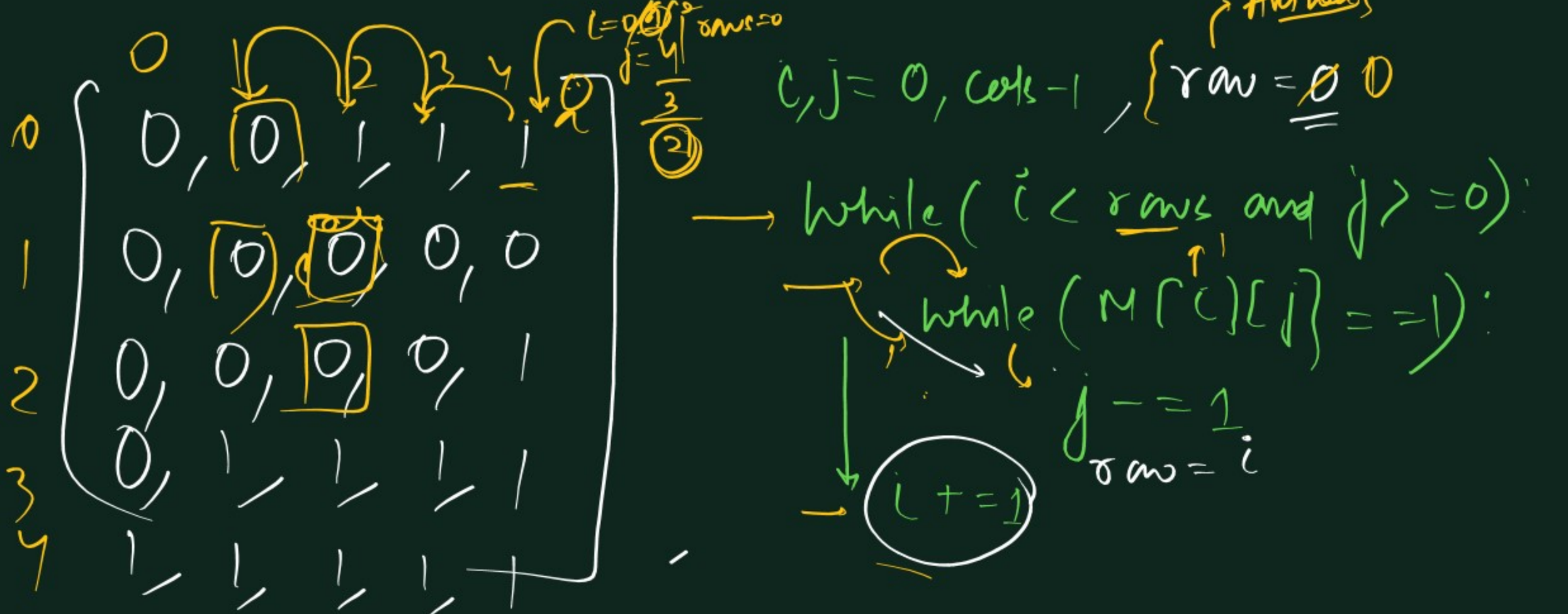
i = [0, — rows-1]

i, j = 0, cols-1

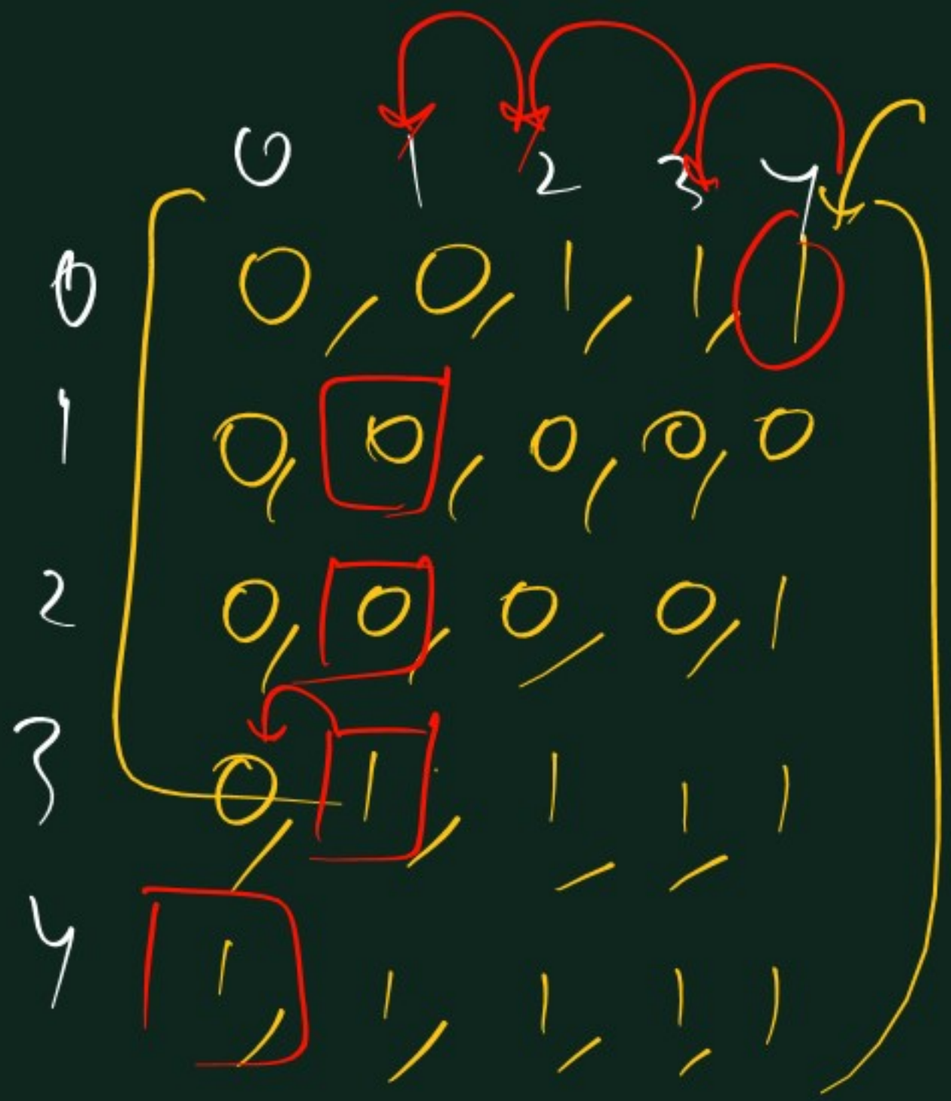
```
while ( i < rows and j >= 0) :
    while ( m[i][j] == 1) :
        j -= 1
    i += 1
return i
```

TC = No of operations | No of ops (m+n)

$O(m+n)$


0 1 2 3 4
0 0 0 1 1 1
0 0 0 0 0
0 0 0 0 1
0 1 1 1 1
1 1 1 1 1
i=0 raws=0
i=4
raws=3
raws=2
c, j = 0, cols-1
{raw = 0 0
Answer
while (i < raws and j >= 0):
while (M[i][j] == 1):
j -= 1
raw = i
i += 1

```
k = -2
while (i < rows and j >= 0):
    if (M[i][j] == 1):
        j -= 1
    elif (M[i][j] == 0):
        i += 1
```



i=0, j=col-1=4

row = 0/3/4

while(i<rows and j>=0):

j>=0 and

while(m[i][j]==1):

j-=1

row=i

i+=1