Introduction to Multidimensional Arrays (2D Arrays)

Problem ①

Masai Maps

Location

Destination

15 Kms → Database

Bangalore → Delhi : 2500

Mumb : 1000

Hyd : 800

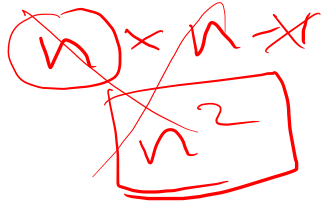
Bang = {
- Delhi : 3000
- Mumb : 2500
- Hyd : 800
}

Delhi = {
Bang
Mumb
Hyd
}

Mumb = {
=
=
}

Hyd = {
=
=
}

4 × 3



Bangalore ⟶ Delhi

Bangalore["Delhi"] = 250

In Objects ⟶ Implementation is easy

But the space is high

Normal Arrays

Bang = [Mumb, Del, Hyd]

Dist = [2500, 3000, 800]

4

Del = [Bang, Mumb, Hyd]

Dist = [

8

$2 \times n \times n$

$2n^2$

Tabulation

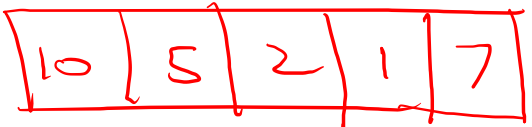
Excel sheet

Name	Maths	Eng	SCI
channu	90	80	75
munnu	35	40	45
pablo pandey	105	120	140
Bablu	100	100	100


B H M D
B 0 800 1500 2500
H 800 0
M 1500 0
D 2500 0

Indexing

Arr =

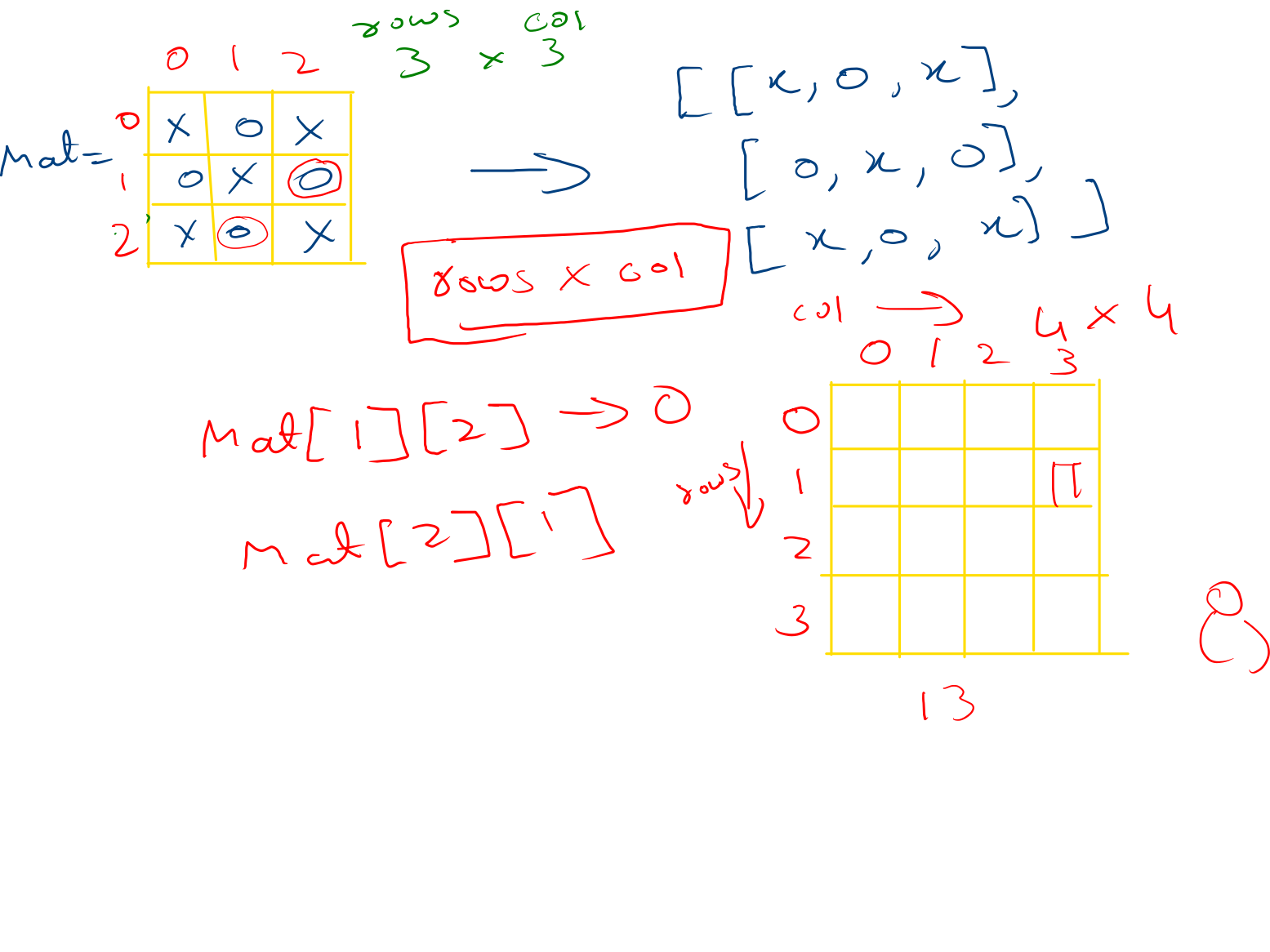


matrix = [[ ], [ ], [ ]]

= [[1, 2, 3], → rows
[4, 5, 6], → rows
[7, 8, 9]] → rows

↑ ↑ ↑

Columns


rows col
3 x 3
0 1 2
Mat=
0 X O X
1 O X O
2 X O X
[[x, o, x],
[o, x, o],
[x, o, x]]
rows x col
col
4 x 4
0 1 2 3
0
1
2
3
rows
Mat[1][2] → O
Mat[2][1]
13

0 1 2 3 4 5 6 7 8 9 10 11 12 13

2D Arrays

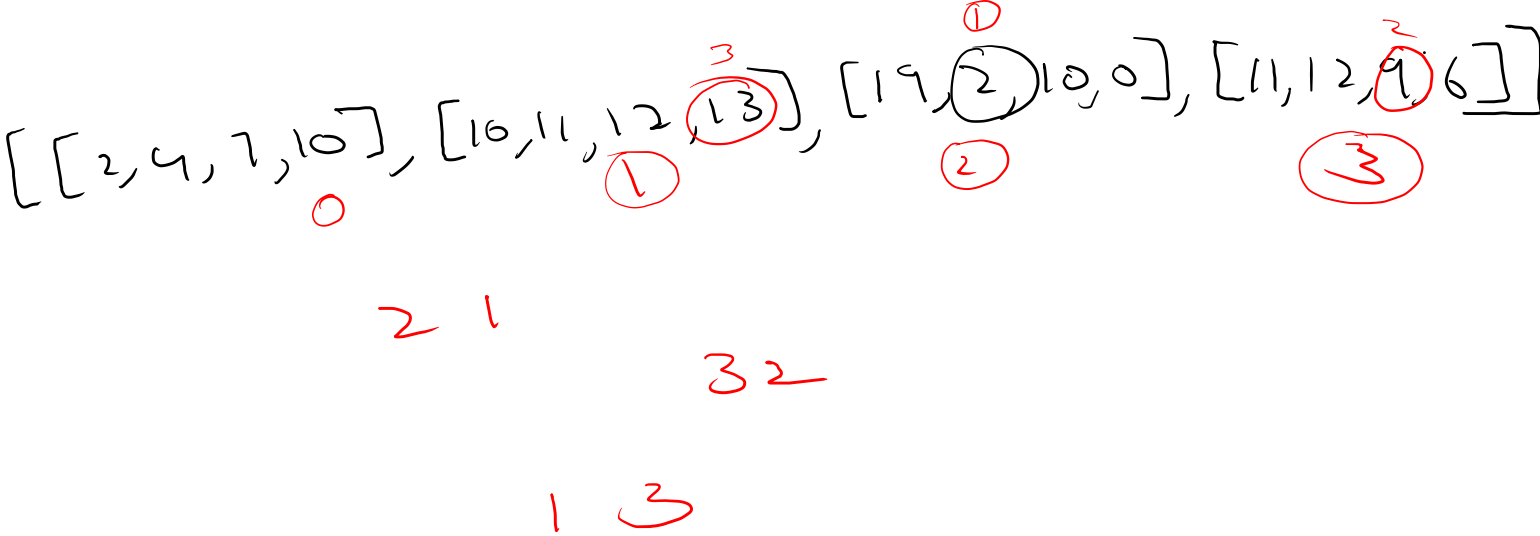
0	L	T	V	L	A	I	A	N	N	C	A	M	N	U
1	R	M	A	P	I	N	U	O	M	T	X	M	I	N
2	A	C	R	A	P	U	L	K	I	T	K	C	U	P
3	H	O	U	B	N	O	C	U	R	T	C	H	L	C
4	U	K	N	L	N	T	O	P	N	M	I	U	O	H
5	L	B	O	O	O	O	U	O	A	Y	B	N	T	O
6	T	C	U	P	B	N	R	B	M	Y	E	N	N	T
7	C	R	N	A	N	R	L	N	N	Y	M	U	N	T
8	U	O	M	N	I	C	D	P	O	C	U	D	H	A
9	I	O	M	D	A	H	U	E	R	A	N	I	H	B
10	O	T	G	E	P	G	R	T	I	A	N	L	R	H
11	C	U	T	Y	M	Y	V	L	P	O	U	N	M	E
12	N	E	N	A	V	O	Y	I	R	T	P	M	L	E
13	B	T	Y	M	Y	T	N	U	A	L	N	I	R	M

~~PULKIT~~ ←
PABLO PANDEY
~~RAHUL~~
~~BICKY~~
~~MUNNU~~
~~VARUN~~
~~CHOTTA BHEEM~~
~~GROOT~~
~~IRON MAN~~
~~CHUNNU~~

2 4 > 2 9

pulkit

→


[[2,4,7,10],[10,11,12,13],[19,2,10,0],[11,12,9,6]]
0
1
2
3
3
1
2
2
1
3
2
1
3

Traverse

3 × 4

	0	1	2	3
0	a	b	c	d
1	e	f	g	h
2	i	j	k	l

arr =

```
let r = 3;
let c = 4;
for(i = 0; i<r; i++){
    for(j = 0; j<c; j++){
        console.log(arr[i][j])
    }
}
```

arr[0][0] a
arr[0][1] b
arr[0][2] c
arr[0][3] d
arr[1][0]

r c

a 0 0

d 1 0

g 2 0

j 3 0

b 0 1

e 1 1

h 2 1

k 3 1

c 0 2

f 1 2

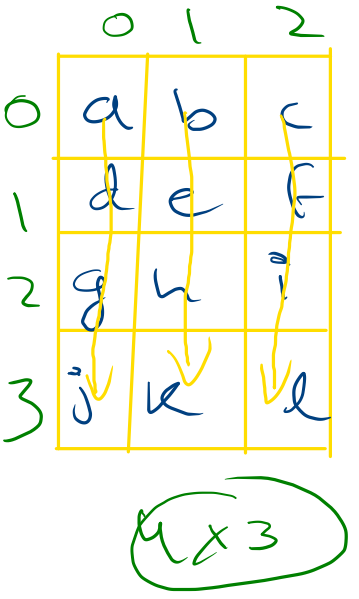
i 2 2

l 3 2

Nested loop

Outer → fixed

Inner → moving




```
let r = 4;
let c = 3;
for(let i = 0; i<c; i++){
      for(let j = 0; j<r; j++){
            console.log(arr[i][j])
      }
}
```

2D Array

1. write down the output → Indexes
2. Which is fixed
   → row or Column
   → outer loop
3. if( )

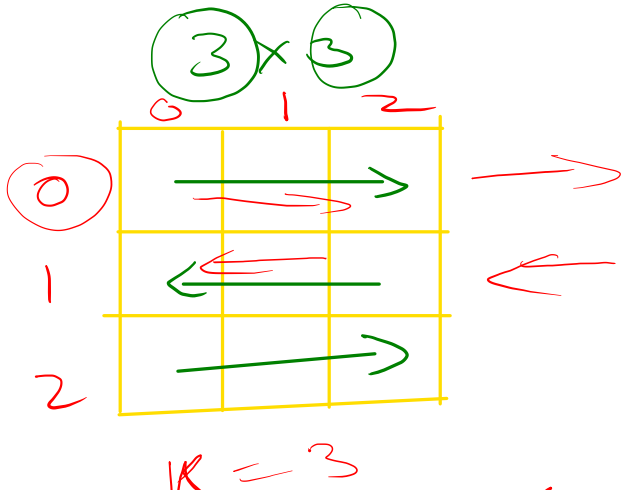
if (i % 2 == 0)

for (→)

else

for (←)

i--



R = 3

row = 3

for ( )

if ( )

for

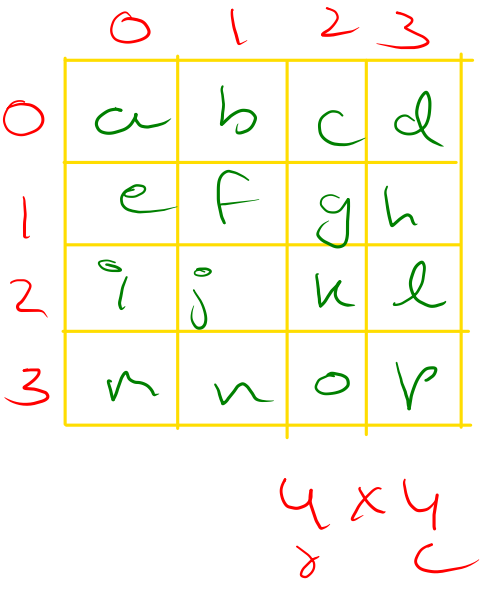
else for

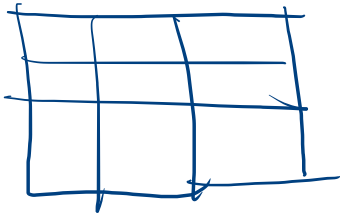
target = "k"

index → 9, 0

```
let r = 4;
let c = 4;

let row = -1; let col = -1;
for(let i = 0; i<r; i++){
      for(let j =0 ;j<c; j++){
            if(arr[i][j] == target){
                  row = i;
                  col = j;
                  break;
            }
      }
}
console.log(row, col);
```





Indexing

Traverse

[[ ],
[ ],
[ ],]