Exit point of a matrix

Starting pt:

source pt - (0,0)

0 - Encounter →

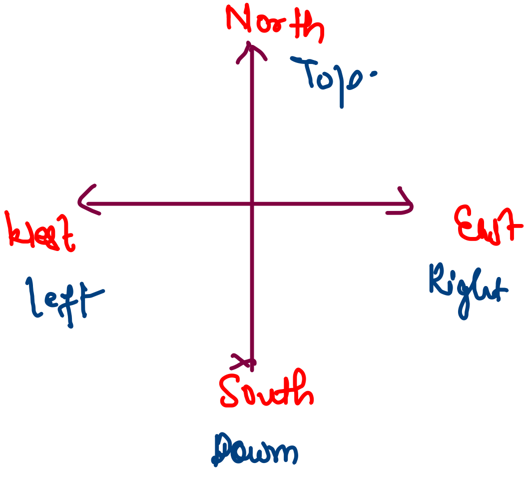
Direction continue.

1. Encounter.

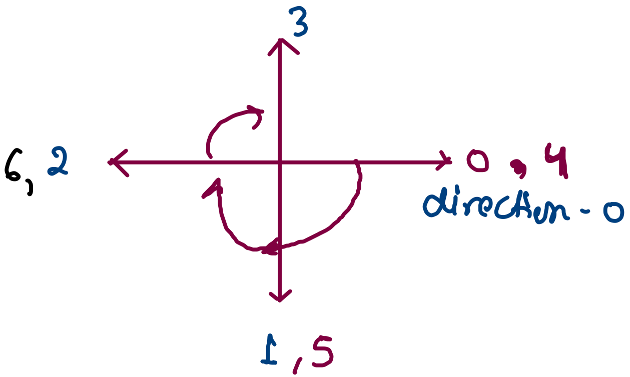
✓ clockwise 90° turn.

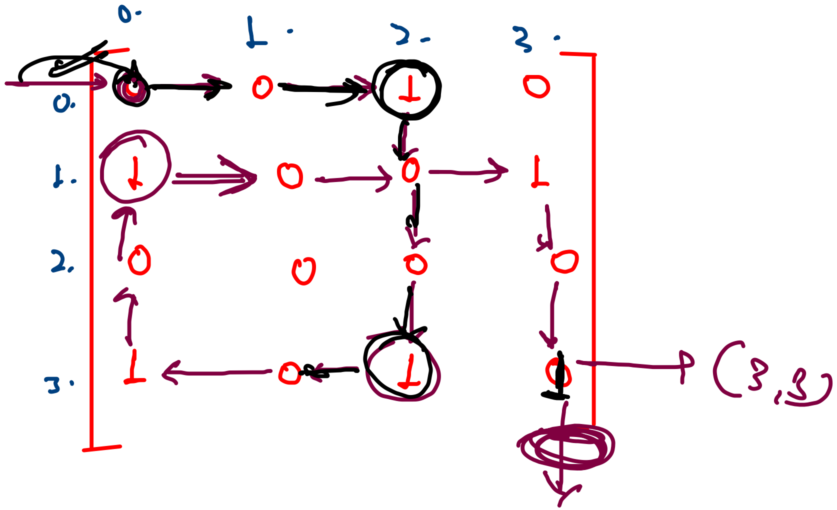
✓ 90° towards Right

$$
\begin{array}{c|cccc}
 & 0 & 1 & 2 & 3 \\
\hline
0 & 0 & 0 & 1 & 0 \\
1 & 1 & 0 & 0 & 0 \\
2 & 0 & 0 & 0 & 0 \\
3 & 1 & 0 & 1 & 0 \\
4 & 0 & 0 & 0 & 0
\end{array}
$$



Exit pt → (1,3)





dir = 0

direction Management →

d = 0 → Right

1 → Down

2 → left

3 → Top

dir = (dir + arr[r][c]) % 4;

0, 1, 2, 3

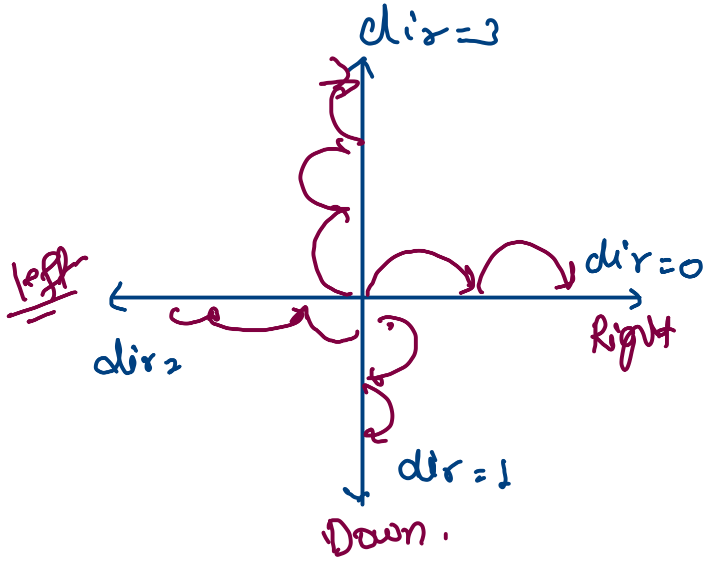
dir = ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~4~~ 5

r = ~~0~~ ~~1~~ 3

c = ~~0~~ ~~2~~ ~~2~~ ~~1~~ 0

(2,3) (1,3)

Moves management



dir = 0

fix → Row

vary → Column ++.

dir = 1

fix → column

vary → Row ++.

dir = 2

fix → Row.

vary → Column --;

dir = 3 fix → Column

move → Row --

r = ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~2~~ 1

c = ~~0~~ ~~1~~ ~~2~~ ~~1~~ ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~4~~ 3

dir = ~~0~~ ~~1~~ ~~2~~ ~~3~~ (4) % = 0

r = ~~2~~ ~~1~~ ~~0~~ -1

c = 0

dir = 3

```
while(r >= 0 && r < arr.length && c >= 0 && c < arr[0].length) {
    // direction management
    dir = (dir + arr[r][c]) % 4;

    // moves management
    if(dir == 0) {
        c++;
    } else if(dir == 1) {
        r++;
    } else if(dir == 2) {
        c--;
    } else {
        r--;
    }
}

System.out.println(r + ", " + c);
}
```

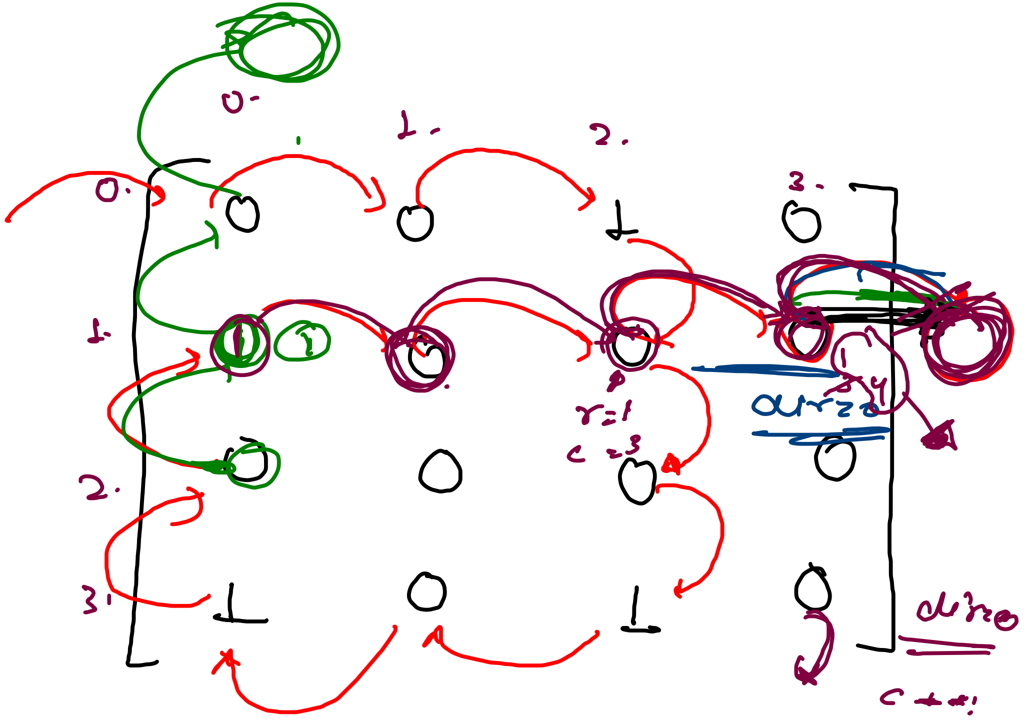
smaller

1, 3

management of invalid zero

dir = 3 ← invalid

r++



point is invalid because of last direction

dir = 0

→ c--

Saddle Point → least price in Row but max price in column.

Row → min
col → max

$$\begin{bmatrix} 11 & 12 & 13 & 14 \\ 21 & 22 & 23 & 24 \\ 33 & 32 & 31 & 34 \\ 40 & 42 & 43 & 44 \end{bmatrix}$$

* Is there any possibility of No. saddle pt? → Yes

Row min

col

* How many Saddle point are possible in arr. → 1 But why??

Saddlepoint

$$\begin{array}{cccc} a & b & c & d \\ e & f\ (p_1) & g & h \\ i & j & k & l\ (p_2) \\ m & n & o & p \end{array}$$

conclusion → p: There is no possibility of more than 1 saddle point

Row → min

Col → max

if 'f' is saddle then,

$f > \{b, j, n\}$

$f < \{e, g, h\}$

$f < h \rightarrow$ (iii)

$f > j \rightarrow$ (iv)

$\boxed{j < f < h}$ combination of (iii) & (iv)

if 'l' is saddle point

$l > \{d, h, p\}$

$l < \{i, j, k\}$

$l < j$ —— (i)

$l > h \rightarrow$ (ii)

$\boxed{h < l < j} \rightarrow$ combination of (i) & (ii)

$j < h$

$h < j$

hence proved

Contradiction.

$$\begin{matrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{matrix}$$

f: p1, l: p2

Row → Min

Col → Max

if 'f' is saddle pt

$f < \{e, g, h\} \longrightarrow f < h$ — (I)

$f > \{b, j, n\} \longrightarrow f > j$ — (II)

combination of (I) & (II)

$f < h, \quad f > j$

→ $\boxed{j < f < h}$

↳ $\boxed{j < h}$

if 'l' is saddle pt.

$l < \{i, j, k\} \longrightarrow l < j$ — (III)

$l > \{d, h, p\} \longrightarrow l > h$ — (IV)

Combination of (III) & (IV)

$l < j, \quad l > h.$

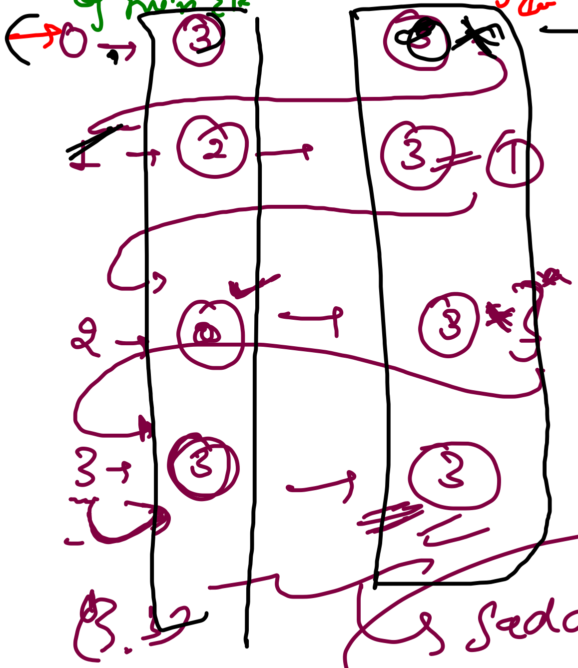
$h < l < j$

↳ $\boxed{h < j}$

using contradiction

find column group of min elem

maxRow snag of this



	0	1	2	3	4
0	14	12	13	11	15
1	23	22	20	24	25
2	31	32	33	34	35
3	44	42	40	41	45

Valid condition

→ Saddle point

r = 3

$\boxed{MnCI = 2}$

maxRI = 3

(r, mnCI)

(r, maxRI) → Saddle point

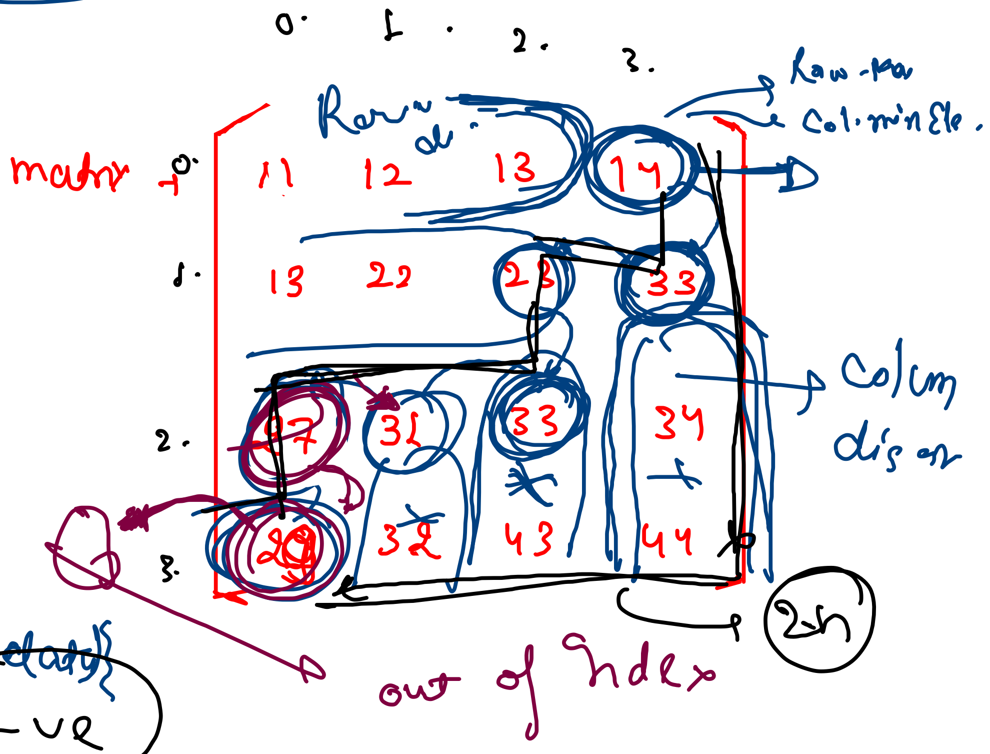
→ Invalid Point

Search In Sorted 2D matrix

data = 29

- → Row wise Sorted
- → Columnwise Sorted
- → Data → 27 → 29

Iteration →




```
if (arr[r][c] == data){
  print(r, c) return } code
} else if (arr[r][c] > data){
  c--;        c -ve
} else if (arr[r][c] < data){
  r++;
}
```

→ Data Not present

r < arr.length
c >= 0

r true

Exit condition from function / function termination

- Return keyword ✓
- all lines are executed ✓