day - 5 Matrix

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

Mtx[3][4] =

Mtx[3][] =

1	2	3		← Mtx[0]
9	5			← Mtx[1]
6	7	8	9	← Mtx[2]

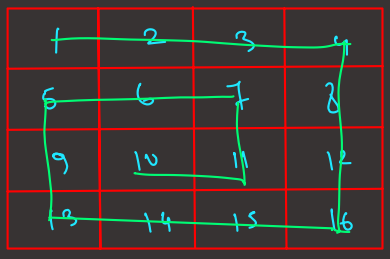
HW

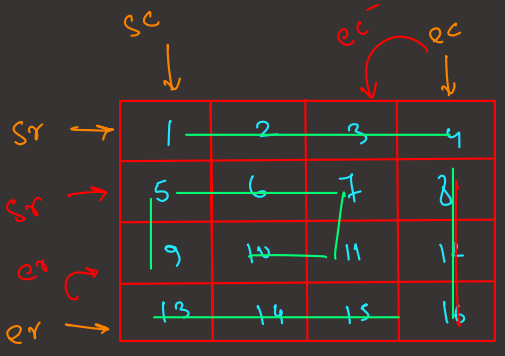
A 3×6 B 6×5

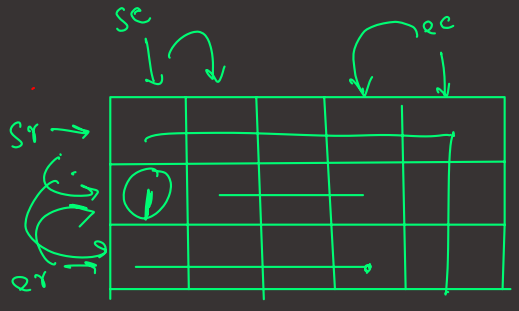
↓

M 3×5

8 Spiral Mtx







Approach

① Top Boundary
② Right Boundary
③ Buttom Boundary
④ Left Boundary.

① Top Boundary
sr → fixed,
print sc to ec.
sr++;
Mtx[sr][i]

② Right Boundary.
ec → fix.
print (sr to er)
ec--
↳ Mtx[i][ec]

③ Buttom Boundary.
er → fix.
print ec to sc
er--
Mtx[er][i]

④ Left Boundary
sc → fixed
print er to sr
sc++;
↳ Mtx[i][sc]

Q Diagonal Sum

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

→ no repeat

n×n

1	2	3
4	5	6
7	8	9

→ repeat

$\left(\frac{n}{2}, \frac{n}{2}\right)$

Diagonal sum = left diagonal sum + Right diagonal sum

```
j = n-1
for (int i = 0 to n) {
    dig.sum = arr[i][i]
            + arr[i][j]
    j--
}
```

Q Search in a sorted Mtx → O(n log n)

sorted arr → O(log n)

→ sorted

↓ sorted

i = 0

n

for (int i = 0; 0;

(Starting position)
(0, m-1)

target = 32

10	20	30	40
15	25	35	45
27	29	37	48
32	33	39	50

ending position
(n-1, 0)

if target < val (pos) ←

if target > value (pos) ↓

Approach (1)

T.C = O(n log m)

row → n

for each row binary search → log m

Approach (2)

→ O(n + m)

row, col

git push -u origin main // for first time

git push

gitHub → local git clone <https link>