2D Array ·

Content

- Find in row wise & column wise sorted matrix
- Row with max no. of 1s
- Spiral Matrix
- Sum of all submatrices sum

Q> Given a row wise and col wise sorted matrix
Find out whether k is present or not.

I/p $A = \begin{bmatrix} -5 & -2 & 1 & 13 \\ -4 & 0 & 3 & 14 \\ -3 & 2 & 6 & 18 \end{bmatrix}$

K = 13 true

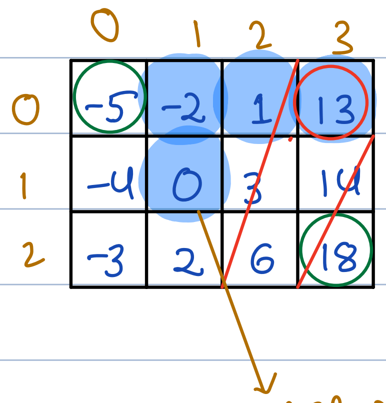
K = 2 true

K = -1 false

Bruteforce

$\forall$ r, c search for element K

R rows C cols → TC : O(R*C)

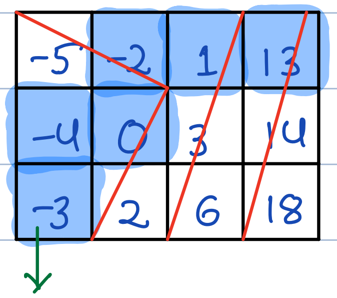


$$\begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & -5 & -2 & 1 & 13 \\ 1 & -4 & 0 & 3 & 14 \\ 2 & -3 & 2 & 6 & 18 \end{array}$$

return true

K = 0

-5 → increasing
↓
increasing



$$\begin{array}{cccc} -5 & -2 & 1 & 13 \\ -4 & 0 & 3 & 14 \\ -3 & 2 & 6 & 18 \end{array}$$

K = -1

NOTE : we can either start from Top Right {13} or
Bottom Left {-3}

Pseudocode {saddle search}

```
R { no. of rows }
C { no. of cols }
A { matrix }.

// Start at Top Right corner.
    r = 0 , c = C-1
```

	0	1	2	3
0	-5	-2	1	13
1	-4	0	3	14
2	-3	2	6	18

```
while ( r<R && c>=0 ) {
        val = A[r][c]
        if (val == k) return true
        if (val > k) {
            c--
        }
        else {
            r++
        }
}

return false
```

TC : O(R+C)

SC : O(1)

Q> Given a binary sorted matrix A[N][N]

Find the row with max # of 1's

NOTE

- If two rows have the max no. of 1, return lower index
- Assume each row to be sorted by values.

I/p

$$A = \begin{array}{c|c|c|c|c} 0 & 0 & 1 & 1 & 2 \\ 1 & 0 & 0 & 1 & 1 \\ 2 & 0 & 1 & 1 & 2 \end{array}$$

output : 0

I/p

$$A = \begin{array}{c|c|c|c|c|c} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 2 & 0 & 0 & 1 & 1 & 2 \\ 3 & 0 & 1 & 1 & 1 & 3 \end{array}$$

output : 3

Bruteforce

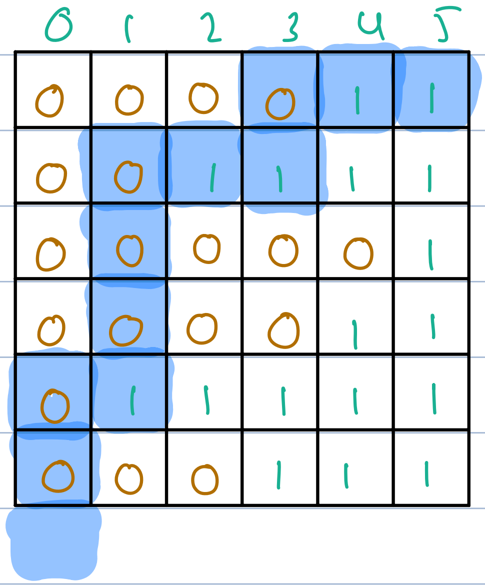
For each row keep track of max # of ones.

TC: O(R*C)

```
mOnes = 0
ans = 0
for r ⟶ 0 to R-1 {
    ones = 0
    for c ⟶ 0 to C-1 {
        if(A[r][c] == 1) ones++
    }
    if(ones > mOnes) {
        ans = r , mOnes = ones
    }
```

}

print (ans)



$$A = \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \\ \hline 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{array}$$

Pseudocode

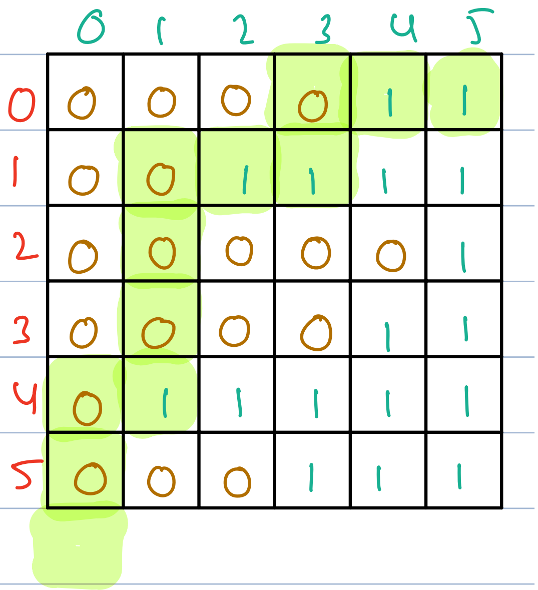
```
r = 0     c = C-1
ans = 0

while ( r < R  &&  c >= 0) {
    while ( c >= 0  &&  A[r][c] == 1) {
        ans = r
        c--
    }
    r++
}
print (ans)
```

TC: O(R+C)

SC: O(1)

Nested for/while loop ≠ R*C

ans = ~~0~~ ~~1~~ 4

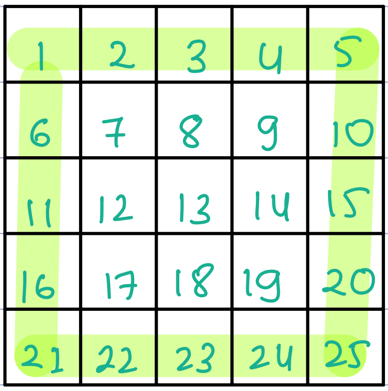


$$\begin{array}{c|cccccc} & 0 & 1 & 2 & 3 & 4 & 5 \\ \hline 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 2 & 0 & 0 & 0 & 0 & 0 & 1 \\ 3 & 0 & 0 & 0 & 0 & 1 & 1 \\ 4 & 0 & 1 & 1 & 1 & 1 & 1 \\ 5 & 0 & 0 & 0 & 1 & 1 & 1 \end{array}$$

Print Boundary Elements ****

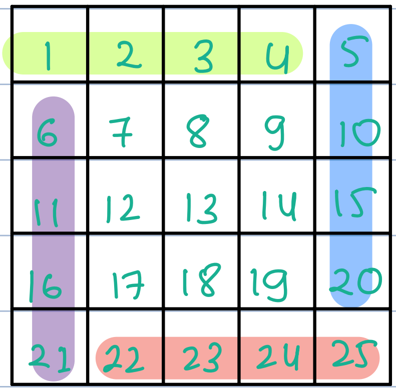
Given a matrix of N * N, Print boundary elements in clockwise direction.

I/p A =



1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

Output : 1 2 3 4 5 10 15 20 25 24 23 22
21 16 11 6



1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

Pseudocode

```
                                                  0     0     N
void printBoundary ( A[N][N], ~~r~~, ~~c~~, ~~N~~ ) {
|
|
|        // Print first row
|
    for k = 1 → N-1 {          // loop N-1 times
        print(A[r][c])
        c += 1
    }

    // Print last col
    for k = 1 → N-1 {          // loop N-1 times
        print (A[r][c])
        r += 1
    }

    for k = 1 → N-1 {          // loop N-1 times
        print(A[r][c])
        c -= 1
    }

    // Print last col
    for k = 1 → N-1 {          // loop N-1 times
        print (A[r][c])
        r -= 1
    }

    if (N == 1) { print(A[r][c]) }
}
```

TC : O(N)

SC : O(1)

Spiral Matrix ****

Given a matrix of N*N.

Print elements in spiral order in clockwise direction.

$$A = \begin{array}{c|cccccc} & 0 & 1 & 2 & 3 & 4 & 5 \\ \hline 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 7 & 8 & 9 & 10 & 11 & 12 \\ 2 & 13 & 14 & 15 & 16 & 17 & 18 \\ 3 & 19 & 20 & 21 & 22 & 23 & 24 \\ 4 & 25 & 26 & 27 & 28 & 29 & 30 \\ 5 & 31 & 32 & 33 & 34 & 35 & 36 \end{array}$$

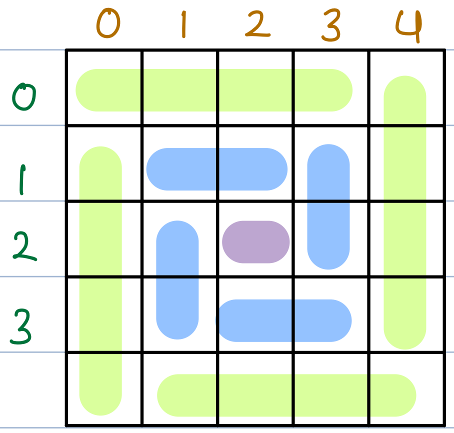
r	c	N
0	0	6
1 +1	1 +1	4 −2
2	2	2

Pseudocode

```
r = 0      c = 0
while ( N>0) {
        printBoundary ( A, r, c, N)
        r += 1
        c += 1
        N -= 2
}
```

TC : $O(N^2)$

SC : $O(1)$

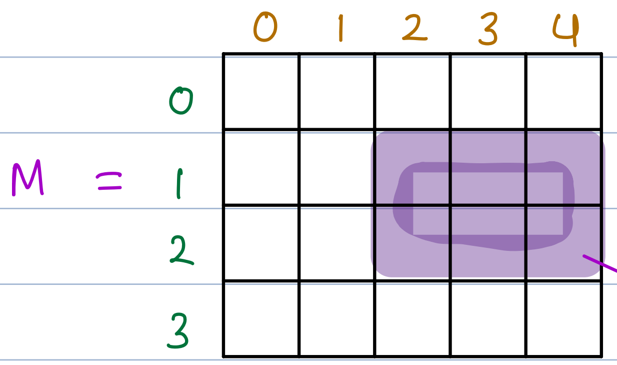


Break 22:45

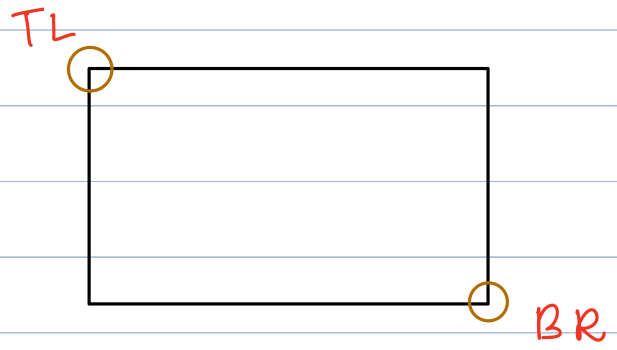
what is a submatrix ?

Subarray → continuous part of an array

Submatrix → continuous part of a matrix → any rectangle in a matrix



TL : (1, 2)
BR : (2, 4)



Sum of all submatrices sum

Given a matrix A[R][C]. Determine the sum of all possible submatrices.

$$M = \begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & 4 & 9 & 6 \\ 1 & 5 & -1 & 2 \end{array}$$

All possible submatrices

size = 1

$\begin{bmatrix} 4 \end{bmatrix}$ 4

$\begin{bmatrix} 9 \end{bmatrix}$ 9

$\begin{bmatrix} 6 \end{bmatrix}$ 6

$\begin{bmatrix} 5 \end{bmatrix}$ 5

$\begin{bmatrix} -1 \end{bmatrix}$ -1

$\begin{bmatrix} 2 \end{bmatrix}$ 2

size = 2

13 $\begin{bmatrix} 4 & 9 \end{bmatrix}$

15 $\begin{bmatrix} 9 & 6 \end{bmatrix}$

4 $\begin{bmatrix} 5 & -1 \end{bmatrix}$

1 $\begin{bmatrix} -1 & 2 \end{bmatrix}$

$\begin{bmatrix} 4 \\ 5 \end{bmatrix}$ 9

$\begin{bmatrix} 9 \\ -1 \end{bmatrix}$ 8

$\begin{bmatrix} 6 \\ 2 \end{bmatrix}$ 8

size = 3

$\begin{bmatrix} 4 & 9 & 6 \end{bmatrix}$ 19

$\begin{bmatrix} 5 & -1 & 2 \end{bmatrix}$ 6

size = 4

$\begin{bmatrix} 4 & 9 \\ 5 & -1 \end{bmatrix}$ 17

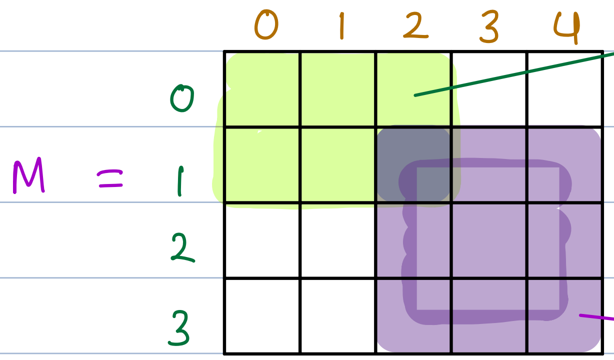
$\begin{bmatrix} 9 & 6 \\ -1 & 2 \end{bmatrix}$ 16

$\begin{bmatrix} 4 & 9 & 6 \\ 5 & -1 & 2 \end{bmatrix}$ 25

Output = 166

Output = 4*(# submatrices 4 is in) → 6

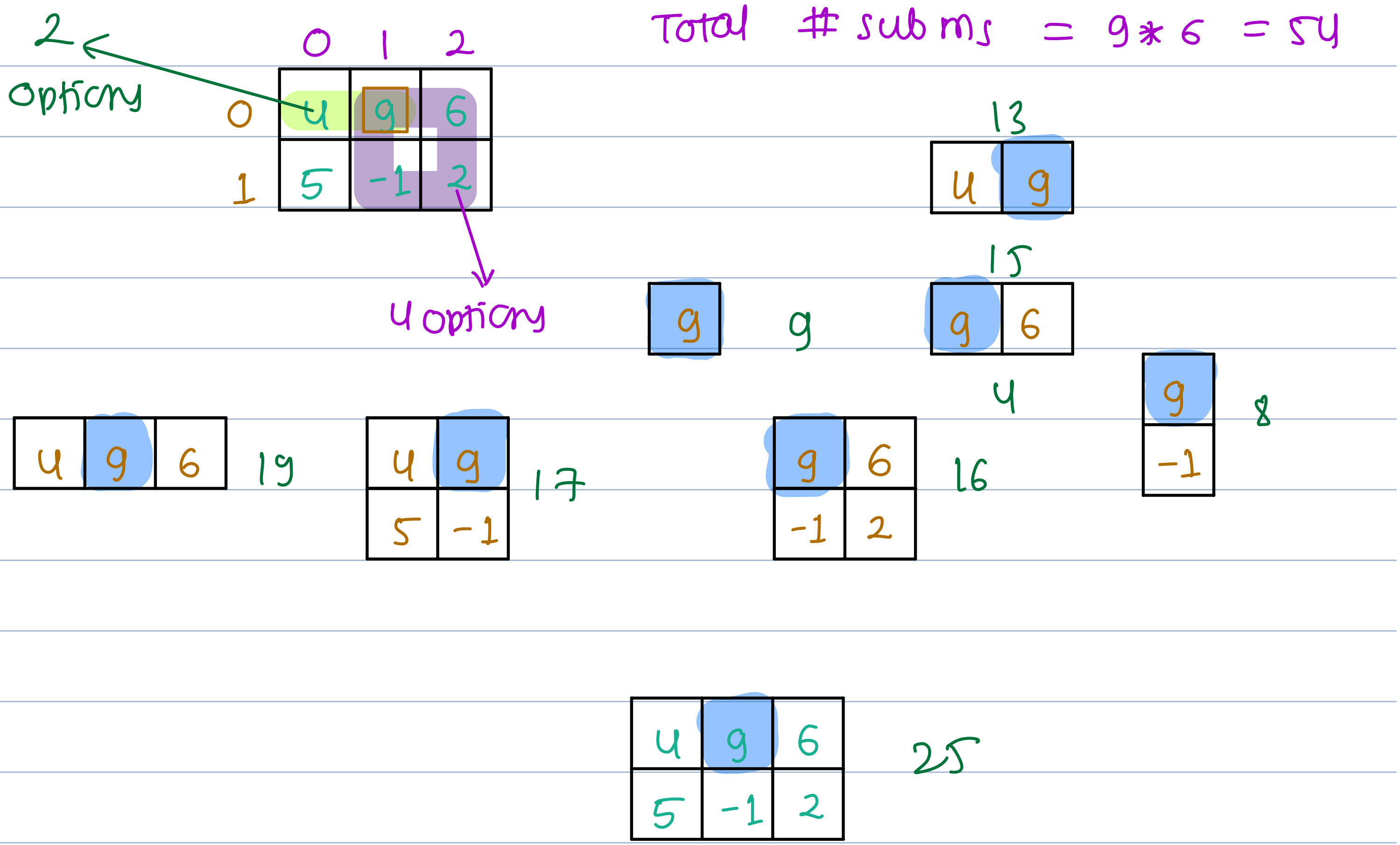
9*(# submatrices 9 is in) → 8 . . . .

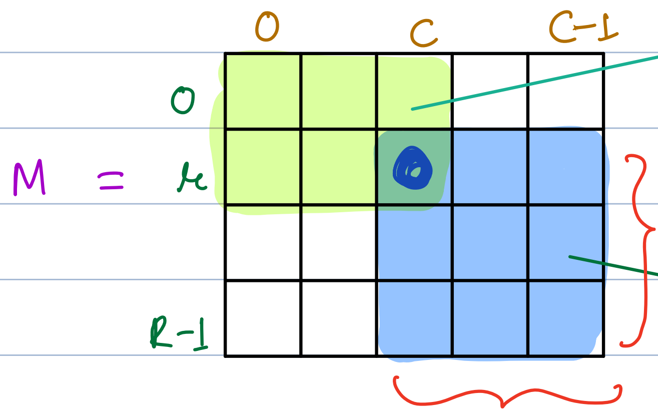


Possible options for TL —— 6

Possible options for BR —— 9

Total # sub ms = 9 * 6 = 54





Possibilities for TL

$(r+1) * (c+1)$

Possibilities of BR

$(C-c) * (R-r)$

$(C-1) - c + 1$

$C - c$

Pseudocode

# submatrices A[r][C] is present in

```
ans = 0

for r ⟶ 0 to R-1 {
    for c ⟶ 0 to C-1 {
        TL = (r+1) * (c+1)
        BR = (R-r) * (C-c)

        ans += (TL * BR) * A[r][c]
    }
}

print(ans)
```

TC : O(R*C)

SC : O(1)

Doubt Session { Attendance is optional }

Brutforce

```
ans = 0
for r1 ⟶ 0 to R-1 {          } TL
    for c1 ⟶ 0 to C-1 {
        for r2 ⟶ r1 to R-1 {      } BR
            for c2 ⟶ c1 to C-1 {
                total = 0
                calculate sum of submatrix
                TL = r1, c1
```

BR = $r_2 C_2$

ans += total

}

TC : $O(R^3 C^3)$

}

}

}

Solve a question

→ Understand the question

↓

Convert input → Ouput

verify

↓ 25 mins to solve the question.

→ getting AC

exceeded 25mins

Hint 1

Hint 2

watch specific part of lecture

Video solution

Revise this Q after a day or week

↓ Till then you still have doubts

Reach out to TA { video call HR }

Post on WA group

DM me on WA

you can see complete sol"

once you have solved the problem

If nothing above helps .

Challenge

PSP — 100%
Att — 100%
Mock interview

End of DSA

1k 2k

Primer questions → 1 primer question to maintain streak .