

Q Given a row-wise & column wise sorted matrix. find whether a given element K is present or not.

M =

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

K = 13 \Rightarrow True

K = 2 \Rightarrow True

K = 15 \Rightarrow False.

Solⁿ

1) Brute force

Iterate over the matrix

T.C. = $O(N \times M)$

S.C. = $O(1)$

2) Optimised Approach

M =

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

K = 2

$$(0, 3) \Rightarrow 13 > 2$$

$\downarrow (j--)$

$$(0, 2) \Rightarrow 1 < 2$$

$\Downarrow (i++)$

$$(1, 2) \Rightarrow 3 > 2$$

$\Downarrow (j--)$

$$(1, 1) \Rightarrow 0 < 2$$

$\Downarrow (i++)$

$$(2, 1) \Rightarrow 2 \text{ (True)}$$

Code

$i = 0, j = M - 1$

while ($i < N$ && $j \geq 0$) {

if ($M[i][j] == K$) {
return True;

}

else if ($M[i][j] < K$) {
 $i++$;

}

else {

j---

T.C. = $O(M+N)$

return false;

Q Given a binary matrix ($M[i][j] = \{0, 1\}$)
Row-wise sorted. ($N \times N$)

Find the row with the max. no. of 1's
In case of multiple rows, return the
one with lower index.

$M = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \rightarrow 0 \text{ (Ans)}$

$M = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} \rightarrow 3 \text{ (Ans)}$

Solⁿ > Brute force

- ⇒ Iterate over every row & count the no. of 1's
- ⇒ Compare the count of each row & take max.

$$T.C. = O(N \times N)$$

	0	1	2	3	4	5	
0	0	0	0	0	1	1	(2)
1	0	0	1	1	1	1	(4)
2	0	0	0	0	0	1	(<4)
3	0	0	0	0	1	1	(<4)
4	0	1	1	1	1	1	(5) ⇒ Ans
5	0	0	0	1	1	1	(<5)

Code

$i = 0, j = N - 1;$

$ans = 0;$

while ($i < N$ && $j > 0$) {

if ($arr[i][j] == 1$) {
 $j--;$
 $ans = i;$

```

    }
    else {
        i++;
    }
    return ans;

```

ans = 4

	0	1	2	3	4	5
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

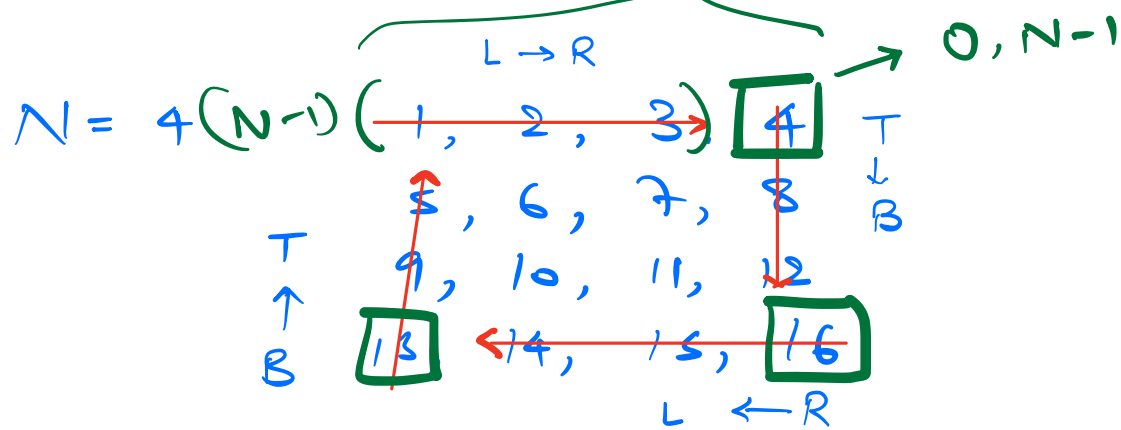
Diagram illustrating a 6x6 matrix with boundary elements highlighted in red. The matrix contains 0s and 1s. The boundary elements (top, bottom, left, and right edges) are marked with red boxes and arrows. The value 4 is written below the matrix, indicating the count of boundary elements that are 1s. A red 'X' is drawn over the bottom-left corner of the matrix.

$$T.C. = O(N+N) = O(N)$$

Given a matrix of size $N \times N$.

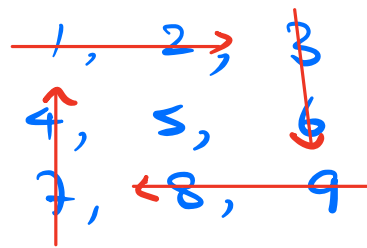
Print the boundary elements in a clock-wise direction starting from (0, 0)

N



O/p : 1, 2, 3, 4, 8, 12, 16, 15, 14, 13, 9, 5

$N = 3$



1, 2, 3, 6, 9, 8, 7, 4

Steps

- 1) Print $N-1$ elements in 1st row ($L \rightarrow R$)
- 2) Print $(N-1)$ elements in last col ($T \rightarrow B$)
- 3) Print $(N-1)$ elements in last row ($R \rightarrow L$)
- 4) Print $N-1$ elements in 1st col ($B \rightarrow T$)

Code

```
void printBoundary ( M[ ][ ], N ) {
```

```
    i = 0, j = 0;
```

```
    for ( count = 0; count < (N-1); count++) {  
        print (M[i][j]);  
        j++;  
    }
```

```
    for ( count = 0; count < (N-1); count++) {  
        print (M[i][j]);  
        i++;  
    }
```

```
    for ( count = 0; count < (N-1); count++) {  
        print (M[i][j]);  
        j--;
```

```
    for ( count = 0; count < (N-1); count++) {  
        print (M[i][j]);  
        i--;
```

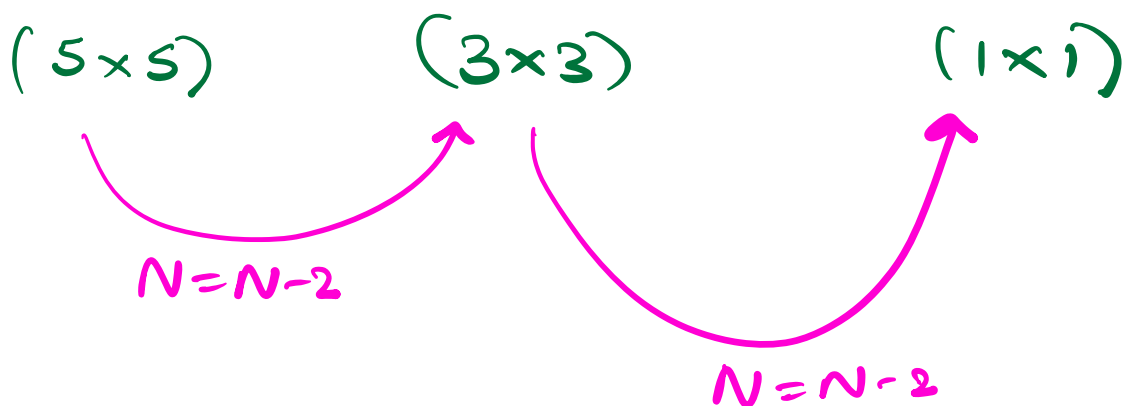
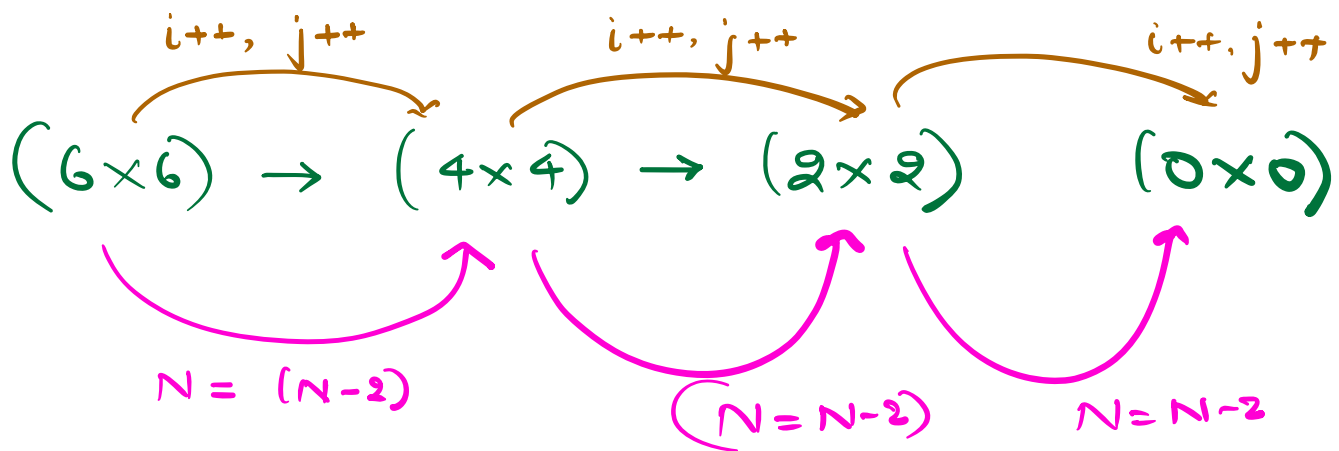
```
}
```

T.C. = $O(4 \times N) = O(N)$

Spiral Order Matrix

$N = 6$

	0	1	2	3	4	5
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



$0 \times 0 \Rightarrow \text{print}(m[i][j])$

$(N > 2)$

Code

```
void printBoundary ( M[][ ], N, i, j ) {
```

```
    for ( count = 0; count < (N-1); count++) {  
        print (M[i][j]);  
        j++;  
    }
```

```
    for ( count = 0; count < (N-1); count++) {  
        print (M[i][j]);  
        i++;  
    }
```

```
    for ( count = 0; count < (N-1); count++) {  
        print (M[i][j]);  
        j--;  
    }
```

```
    for ( count = 0; count < (N-1); count++) {  
        print (M[i][j]);  
        i--;  
    }
```

$K = N;$

$i = 0, j = 0;$

while ($K > 2$) {

 printBoundary (M, K, i, j);

$K = K - 2;$

$i++;$

$j++;$

}

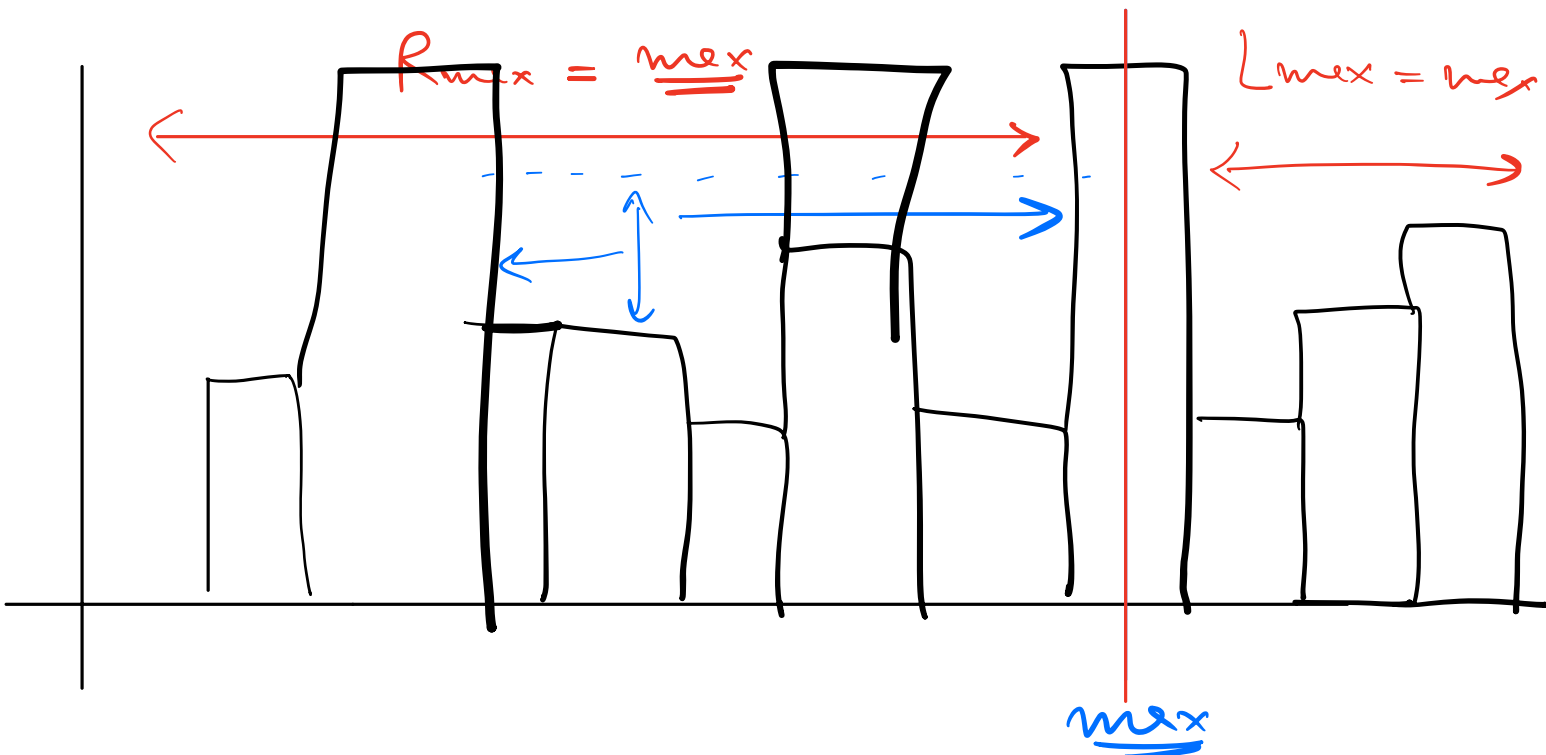
if

($K == 1$) {

 print ($M[i][j]$);

}

T.C. = $O(N^2)$
S.C. = $O(1)$

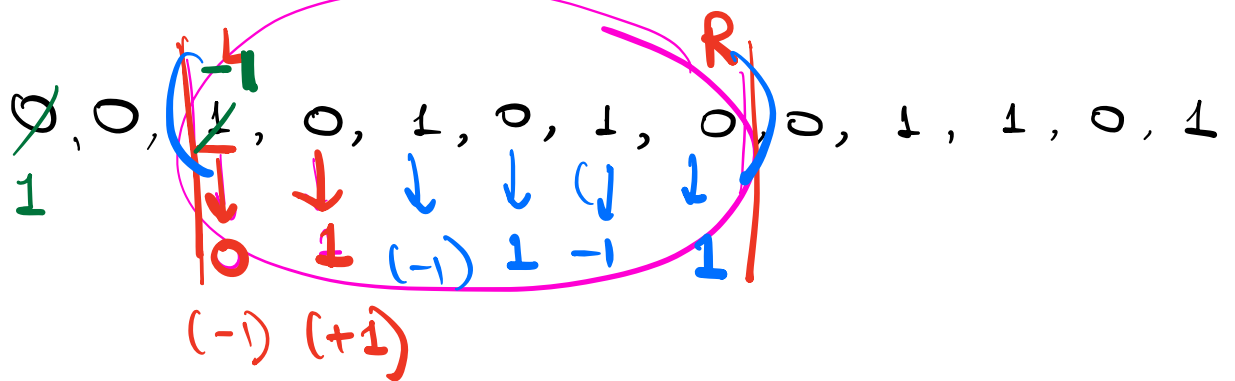


Reattempt 2 ↩

Sat (6:00 AM)



Sund (12:00 PM)



max subarray sum