7 days.

Total ⇒ 168

Sleep ⇒ 49 (7×7)

Office ⇒ 50 (5×10)

Normals ⇒ 21 (7×3)

Fun ⇒ 21 (7×3)

Scaler class ⇒ 9 (3×3)

⇒ 150

Prac ⇒ 168 − 150 ⇒ 18

Q1 Given an array of size N. Find the maximum value of $j-i$ such that $A[i] \leq A[j]$.

Constraints: $1 \leq n \leq 10^5$ ⇒ TC possible ⇒ $O(n)$, $O(n \log n)$, $O(n\sqrt{n})$

$-10^9 \leq A[i] \leq 10^9$

Eg : $A = [3, 5, 4, 2]$ (indices 0 1 2 3)

$j = 3$: no i available

$j = 2$: $i = 0$, ans ⇒ $2-0$ ⇒ 2

$j = 1$: $i = 0$, ans ⇒ $1-0$ ⇒ 1

$j = 0$: no j available

Brute Force: Check for every pair.

TC: $O(n^2)$.

SC: $O(1)$.

Hint 1:

(first, second)
↓ value ↓ index

A = [3, 5, 4, 2]
0 1 2 3

⇒ [(3,0), (5,1), (4,2), (2,3)] → sort on first term → [(2,3), (3,0), (4,2), (5,1)]

arr[i]

arr[j]

0	1	2	3
3	0	0	0

j = 1
i = 0 = 1

j = 2 ⇒ 2
i = 0

left min[n]

left-min[i] ⇒ min element encountered in arr[0 ... i]

indexes

0	1	2	3
3	0	2	1

-1

0	1	2	3
2	3	4	5

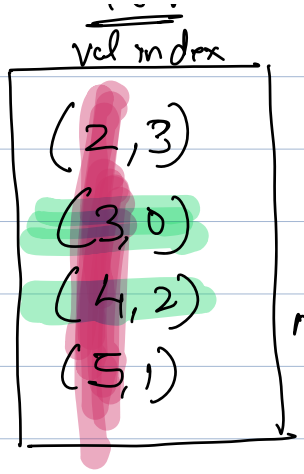
⇒

0	1	2	3
3	0	0	0

j = 1

↳ left min[1.2]

arr arr[0] will never be our i



min_index ⇒ arr[0].index ⇒ 3

j = 0 { arr[i].index }

min_index = 0 as j - i ⇒ -3

j = 2 { arr[2].index }

min_index = 0 as ⇒ 2 - 0 ⇒ 2

j = 1 { arr[3].index }

as ⇒ 1 - 0 ⇒ 1

min_index ⇒ min { min_index, arr[i].index }

[1 2 3 4 5]

0 1 2 3 4

-1

((1, 0) (2, 1) (3, 2) (4, 3) (5, 4)) ⇒ 4

[5, 4, 3, 2, 1]

0 1 2 3 4

j i

A[j] ≥ A[i]

j-1

(1,4)

(2,3) ⇒ -1

(3,2) ⇒ -1

(4,1)

(5,0)

TC: O(n log n)

arr, ArrayList

```
vector< pair <int, int>> vec;

vec.push_back (make_pair (arr[i], i));
```

Q2 Given a matrix of size N x M, find the sum of all submatrix sum.

Ans

	0	1
0	3	1
1	-1	-2
2	2	4

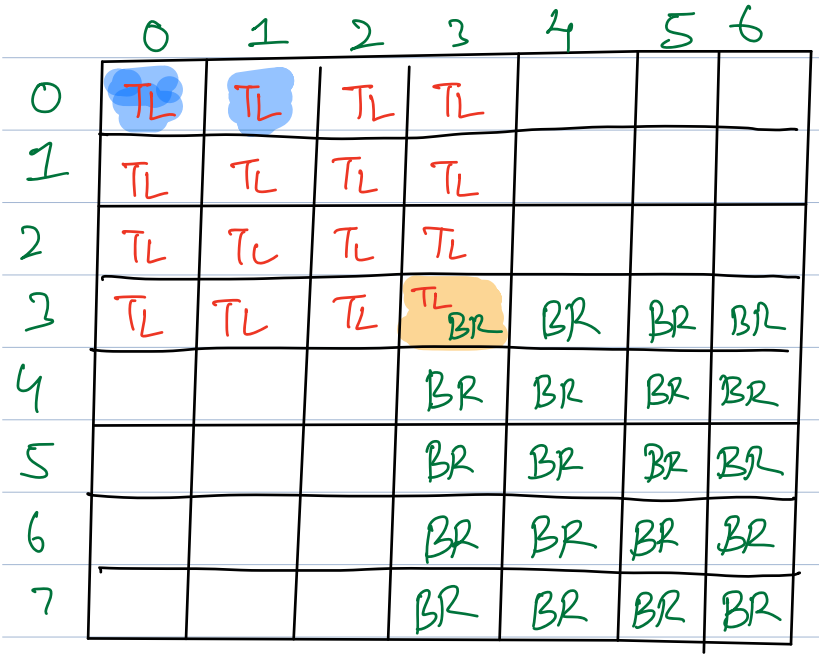
$\Rightarrow$

$$\begin{bmatrix} [3], & [3,1] & \begin{bmatrix}3\\-1\end{bmatrix} & \begin{bmatrix}3&1\\-1&-2\end{bmatrix} \\ \begin{bmatrix}3\\-1\\2\end{bmatrix} & \begin{bmatrix}3&1\\-1&-2\\2&4\end{bmatrix} & [1], & \begin{bmatrix}1\\-2\end{bmatrix} \\ \begin{bmatrix}1\\-2\\4\end{bmatrix} & [-1] & [-1,-2] & \begin{bmatrix}-1\\2\end{bmatrix} \\ \begin{bmatrix}-1&-2\\2&4\end{bmatrix} & [-2] & \begin{bmatrix}-2\\4\end{bmatrix}, & [2], [2,4] \\ & & & \begin{bmatrix}4\end{bmatrix} \end{bmatrix}$$

$$\text{Sum} \Rightarrow (3\times6)+(1\times6)+(-1\times8)+(-2\times8) + (2\times6)+(4\times6) \Rightarrow \boxed{36}$$

$[a,b] \Rightarrow b-a+1$

$[j, n-1] \Rightarrow n-1-j+1 = (n-j)$



$(i,j) \Rightarrow (3,2)$

no of TL $\Rightarrow (i+1) \times (j+1)$

no of BR $\Rightarrow (n-i) \times (m-j)$

$20 + 20 + 20 + 20 + 20$

$\Rightarrow$ 16 times

$16 \times 20 \Rightarrow \boxed{320}$

Break: $\boxed{10:45}$

Q3 Given a 2D Matrix, every row sorted, every column sorted. Find an element K.

n = rows
m = columns

K=15

	0	1	2	3	4	5
0	-1	2	4	5	9	11
1	1	4	7	8	10	14
2	3	7	9	10	12	18
3	6	10	12	14	16	20
4	9	13	16	19	22	24
5	11	15	19	21	24	27
6	14	20	25	29	31	39
7	18	24	29	32	34	42

Brute force 1: Iterate on every element: TC: $O(mn)$

Brute force 2: Iterate on each row and do binary search TC: $O(n \log m)$

Brute force 3: Iterate on each column and do binary search TC: $O(m \log n)$

K=15

	0	1	2	3	4	5
0	-1	2	4	5	9	11
1	1	4	7	8	10	14
2	3	7	9	10	12	18
3	6	10	12	14	16	20
4	9	13	16	19	22	24
5	11	15	19	21	24	27
6	14	20	25	29	31	39
7	18	24	29	32	34	42

```
i = 0 , j = m-1

while ( j >= 0 && i < n ) {
    if (arr[i][j] > k) {
        // ignore below
        j--;
    } else if (arr[i][j] < k) {
        // ignore left
        i++;
    } else
        return true;
}
return false;
```

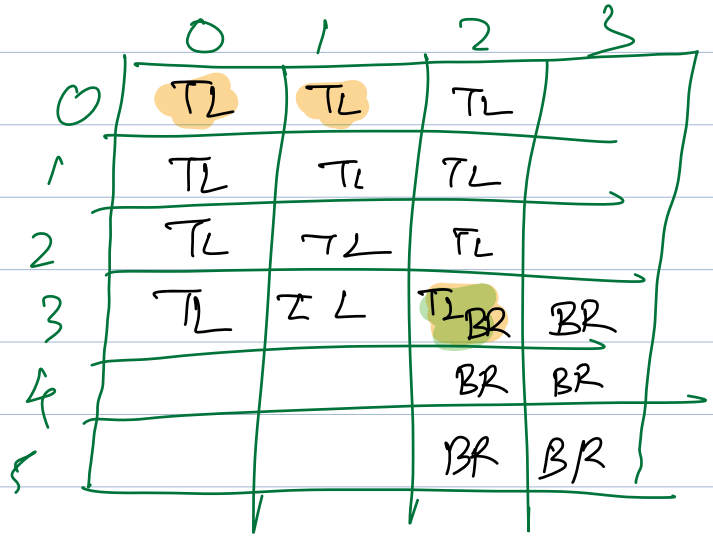
TC : O(m+n)
SC : O(1)

i+1 = A , j-- = B

A, A, B, B, A, B, A, B ....

(n + m) ⇒

6×4



TL = 12

BR ⇒ 6

6 + 6 + 6 + 6 + 6

12 × 6 ⇒ 72

72 × a[0][0]