Q Given a matrix with all rows & cols sorted in desc order. Find the max sum of any submatrix.

$$\checkmark \begin{array}{|cc|c} \hline 8 & 5 & 2 \\ 4 & 2 & -3 \\ \hline -1 & -4 & -6 \end{array} \Rightarrow 19 \qquad \checkmark \begin{array}{|c|c} \hline -1 & -2 \\ \hline -3 & -4 \end{array} \Rightarrow -1$$

$$\begin{array}{|ccc|} \hline 7 & 6 & 2 \\ 4 & 3 & 1 \\ 3 & 2 & -1 \\ \hline \end{array} \Rightarrow 27 \qquad \begin{array}{|cc|} \hline 3 & 1 \\ 2 & 1 \\ \hline \end{array} \Rightarrow 7$$

Bruteforce

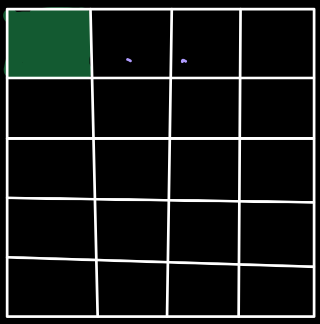
⇒ // Iterate over all submatrix $\Rightarrow O(N^4)$

⇒ // For every submatrix iterate & get the sum $\Rightarrow O(N^2)$

$$\begin{tikzcd} TC: O(N^6) \arrow[d, "\text{After PS Matrix}"] \\ O(N^4) \end{tikzcd}$$

Obs

- The element (0, 0) is the max element.
- Hence (0, 0) will always be the part of the ans.
- TL of max sum subMatrix will be (0, 0)
- Find the BR corner for TL (0, 0)



TL : $(\overset{i}{0}, \overset{j}{0})$

```
for(p=0; p<N; p++){          } O(N^2)
    for(q=0; q<N; q++){
        // BR -> p, q

        // find sum from (0,0) -> (p, q)
        sum=0
        for(r=i; r<=p; r++){          } O(N^2)
            for(c=j; c<=q; c++){
                sum += A[r][c];
            }
        }
    }
}
```

PS Matrix
O(1)

TC : $O(N^2)$ + $O(N^2)$ ⟶ $O(N^2)$

↓ Build PS

↓ Finding the BR

➔ PS[i][j] ⟶ Sum of submatrix from 0, 0 → i, j

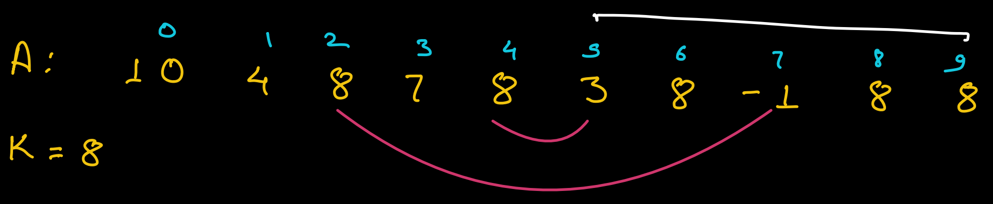
⇒ While building PS matrix keep updating MaxSum

↓ Ans

TC : $O(N^2)$

Amazon
Linked In

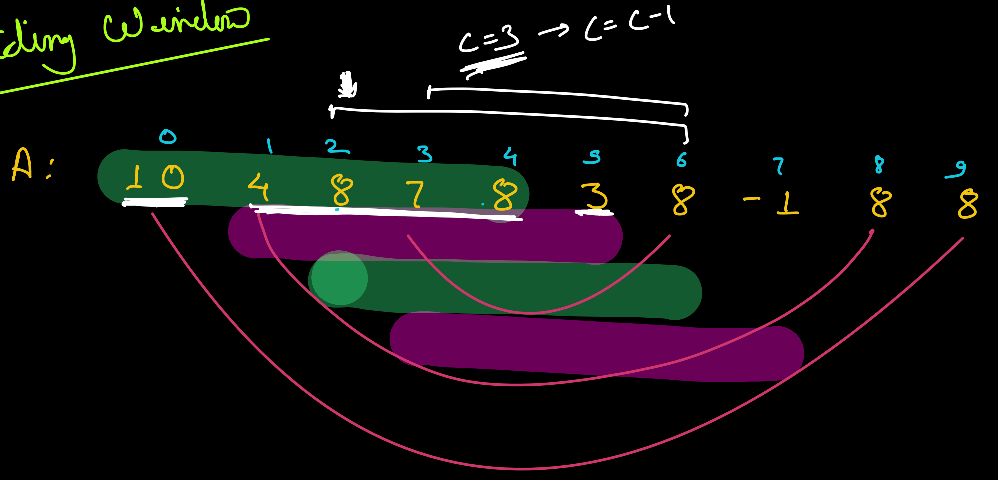
Q Given an array & a no x.
Count the min no of swaps required to bring all element equal to x together.



K = 8

Ans → 2

Sliding Window



K=5	No of 8s x	No of swaps
[0, 4] →	2	(5-2) ⇒ 3
[1, 5] →	2	(5-2) ⇒ 3
[2, 6] →	3	(5-3) ⇒ 2
[3, 7] →	2	(5-2) ⇒ 3
[4, 8] →	3	(5-3) ⇒ 2
[5, 9] →	3	(5-3) ⇒ 2

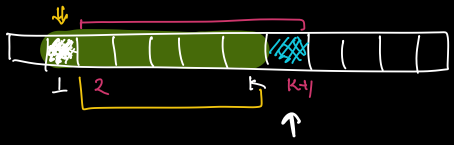
final ans = 2

K ⇒ Count of x in array

C_1 1st window of size K : $[0, K-1]$ ⇒ iterate over window to count no of x's $\rightarrow \boxed{K}$

C_2 2nd window : $[1, K]$ $\Rightarrow \overset{O(1)}{C_1} + \begin{cases} -1 & \text{if } A[0]==x \\ +1 & \text{if } A[K]==x \end{cases}$

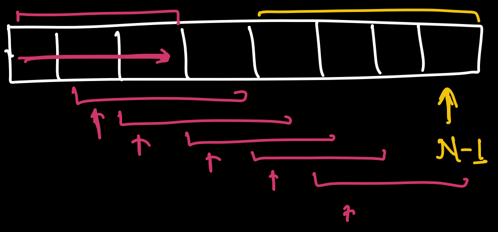
C_3 3rd window : $[2, K+1]$ $\Rightarrow \overset{O(1)}{C_2} + \begin{cases} -1 & \text{if } A[1]==x \\ +1 & \text{if } A[K+1]==x \end{cases}$



C_4 4th window : $[3, K+2]$ $\Rightarrow \overset{O(1)}{C_3} + \begin{cases} -1 & \text{if } A[2]==x \\ +1 & \text{if } A[K+2]= \end{cases}$

⋮

last window : $[N-K, N-1]$ $\Rightarrow \overset{O(1)}{C_{N-1}} + \begin{cases} -1 & \text{if } A[N-K-1]==x \\ +1 & \text{if } A[N-1]==x \end{cases}$



$\begin{bmatrix} [x, N-1] = K \\ x \Rightarrow ? \end{bmatrix}$

$K + N - K \Rightarrow \underline{O(N)}$

Steps I

Build a window of size K

- Keep count of x from index 0 to K-1

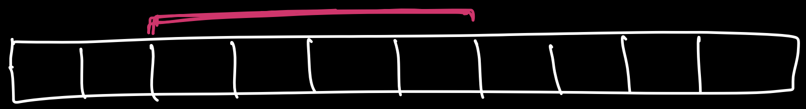
  # iterations = K

Step II

Iterate over remaining windows.

- Remove contribution of 1st element of prev window
- Add contribution of last element of curr wind.

  # iteratn = N-K



freq(x) = K

Break till 10:25 p

Q Given a matrix find the maximum sum submatrix such that the sub matrix [starts from row = 0
ends at row = N-1

A (3x5) →

-3	4	2	2	9
-9	-3	3	3	-3
-1	6	-4	4	-10

⇒ Since Cols will not be considered partially → Create 1d Col Sum array

Kadane's Algo ←

-13	7	1	9	-4

⇒ Find Max Sum subarray in Col Sum (1d) arr $O(N)$

No of valid sub matrix starting from Col 0 ⇒ M
+
No of valid sub matrix starting from Col 1 ⇒ M - 1
+
No of valid sub matrix starting from Col 2 ⇒ M - 2
⋮
+
No of valid sub matrix starting from Col M-1 ⇒ 1

$$\frac{M(M+1)}{2}$$

⇒ $O(M^2)$

TC : $O(M*N) + O(M) \Rightarrow O(MN)$

↓ Build Col Sum

↑ Kadane's Algo

Q Given a matrix find the maximum sum submatrix such that the submatrix starts from $\boxed{row = 0}$ (end anywhere)

1	3	-10	-3	4
-3	-4	2	-2	10
2	6	-9	6	-1
-4	2	4	-3	-8

Ans can have

start	end
0	0 ⇒
0	1
0	2
0	3
⋮	⋮
0	N-1

Max Sum SubMatrix

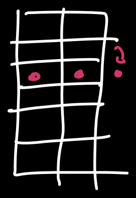
start	end						Max Subarray sum	Max Sum
0	0	1	3	-10	-3	4	4	4
0	1	-2	-1	-8	-5	14	14	14
0	2	0	5	-17	1	13	14	14
0	3	-4	7	-13	-2	5	7	14

TC : $O(N*M) + O(M*N) \Rightarrow O(NM)$

↑ Col Prefix Sum

N time Kadanes On M size array

100x 2



```
ColPS[M] = {0}
StatRow = 0,

for ( endRow = 0; endRow < N; endRow ++) {

        for ( j = 0; j < M; j ++ ) {
            ColPS [j] += Mat [ endRow] [j] ;
        }

        Max SubArray Sum = Kadane ( ColPS );
        maxSum = max (Max SubArray Sum , maxSum);
}
```

Google
Facebook
Direct-i
Amazon
LinkedIn
⋮

Q Find the max subMatrix sum.

0, -2, -7, 0

9, 2, -6, 2

-4, 1, -4, 1

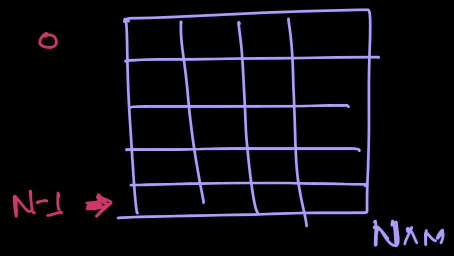
-1, 8, 0, -2

⇒ 15

Possible ans matrix can have

Start	end
0	0
0	1
⋮	⋮
0	N-1
1	1
1	2
⋮	⋮
1	N-1
2	2
2	3
⋮	⋮
2	N-1
3	3
3	4
⋮	⋮
3	N-1
⋮	⋮
N-1	N-1
M-1	N-1

⇒ Prev part



Hashing 2 ⇒ ALGO (Sliding Window)

```
for ( startRow = 0;  startRow < N;  startRow ++) {
  ⇒ ColPS[M] = {0}
    for (endRow = startRow; endRow < N;  endRow ++) {

        for ( j = 0;  j < M;  j++ ) {
              ColPS [j] + = Mat [ endRow] [j] ;
        }

        MaxSubarrSum  =  Kadane ( ColPS );
        maxSum        =  max (MaxSubarrSum , maxSum);
    }
}
```

N^2 M^2

M N

TC : $O(N^2 M)$ SC : $O(M)$

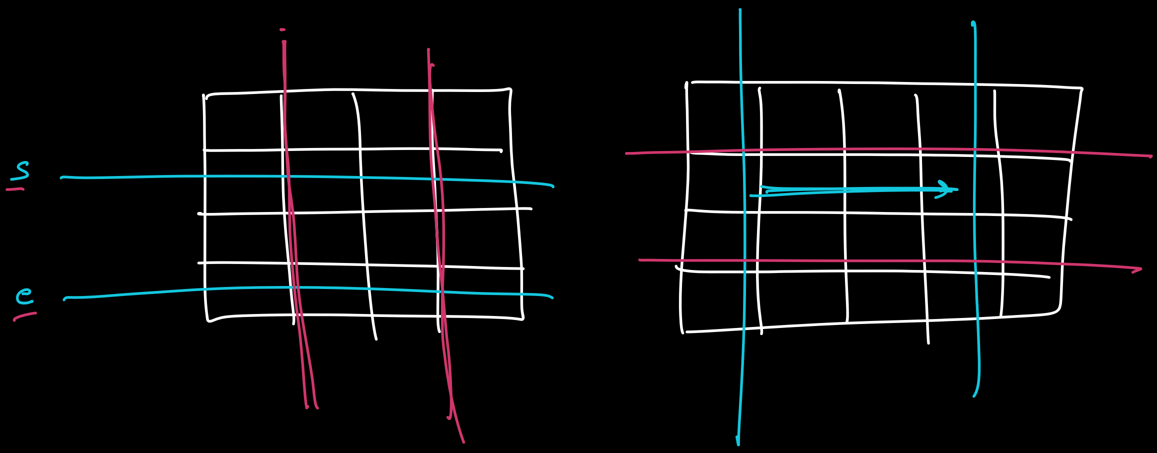
if N >> M is given in problem constraint

→ Take row PS

→ Apply Kadane on Row PS → TC : $O(N)$

→ TC : $O(M^2 N)$

→ SC : $O(N)$


s
e