Q. Given an array [N]. Find total sum of all subarrays.

Ex = [1, 2, 3]

[1]	=	1
[1, 2]	=	3
[1, 2, 3]	=	6
[2]	=	2
[2, 3]	=	5
[3]	=	3
		20

ans = 20

BF : Every subarray, get the sum. and add it tot sum

TC: $O(N^2 \cdot N)$

SC: $O(1)$

PF sum

TC: $O(N^2)$

SC: $O(N)$

carry forward

TC: $O(N^2)$

SC: $O(1)$

// construct PF[]

arr: [1, 2, 3]

PF[]: [1, 3, 6]

$$arr[\,] = [\overset{0}{1}, \overset{1}{2}, \overset{2}{3}]$$

tot = ~~1~~ ~~4~~ ~~10~~ ~~12~~ ~~17~~ 20

```
totsum = 0
for ( i = 0; i < N; i++ )
{
    for ( j = i; j < N; j++ )
    {
        // (i-j) subarray

        if ( i != 0 )   sum = PF[j] - PF[i-1]
        else            sum = PF[j]

        totsum += sum
    }
}
return totsum
```

TC: $O(\underbrace{N}_{\text{Construct PF}} + N^2) : O(N^2)$

SC: $O(\underbrace{N}_{PF[\,]})$

$$[\overset{0}{1}, \overset{1}{2}, \overset{2}{3}]$$

K = 1

K = 1, 2

K = 1, 2, 3

```
sum = 0
for ( j = 0; j < N; j++ )
{
    sum += a[j]
    // sum -> sum[0, j]
}
```

k = [2]

k = [2, 3]

```
sum=0
for( j=1 ; j<N ; j++)
{
    sum += a[j]
    // sum -> sum [1, j]
}
```

```
sum=0
for( j=2 ; j<N ; j++)
{
    sum += a[j]
    // sum -> sum [2, j]
}
```

```
totsum = 0

for ( st=0 ; st<N ; st++)
{
    sum=0
    for( j=st ; j<N ; j++)
    {
        sum += a[j]
        // sum -> sum [st, j]
        totsum += sum
    }
}
```

~~sum~~ totsum

TC: $O(N^2)$

SC: $O(1)$

$\begin{matrix} {\scriptstyle 0} & {\scriptstyle 1} & {\scriptstyle 2} \\ [\,1, & 2, & 3\,] \end{matrix}$

[1]	= 1	= 1	
[1, 2]	= 1 + 2	= 3	
[1, 2, 3]	= 1 + 2 + 3	= 6	
[2]	= 2	= 2	
[2, 3]	= 2 + 3	= 5	
[3]	= 3	= 3	
		20	

$= arr[0] \cdot 3 + arr[1] \cdot 4 + arr[2] \cdot 3$

$= 1 \cdot 3 + 2 \cdot 4 + 3 \cdot 3$

$= 3 + 8 + 9$

$= 20$

arr [0, N-1]

Sum of all subarray = arr [0] × No of subarrays contains arr [0] + arr [1] × No of subarrays contains arr [1]

+ arr [2] × No of subarrays contains arr [2] + ... N - 1

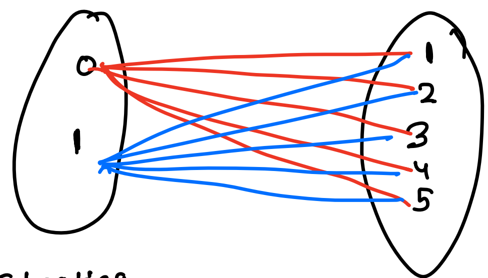
Q No of subarrays having A[1] in them

$$\begin{array}{lcccccc} & 0 & 1 & 2 & 3 & 4 & 5 \\ arr[\,] = & [3, & -2, & 4, & -1, & 2, & 6] \end{array}$$

ans = 10

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

→ ans = 10



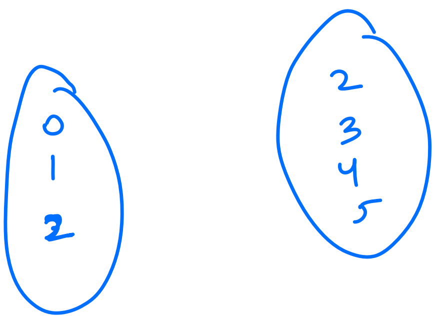
starting

ending

Q arr[2] in them

$$\begin{array}{lcccccc} & 0 & 1 & 2 & 3 & 4 & 5 \\ arr[\,] = & [3, & -2, & \boxed{4}, & -1, & 2, & 6] \end{array}$$

[0 2]	[1-2]	[2-2]
[0 3]	[1-3]	[2-3]
[0 4]	[1-4]	[2-4]
[0 5]	[1-5]	[2-5]



3 × 4 = 12

Q. How many subarray contains arr[i]

0
1
2
3
⋮
i

$i+1$

i
i+1
i+2
⋮
N-1

$(N-1)-(i)+1$

$= N-i$

$= (i+1)(N-i)$

arr[0, N-1]

Sum of all subarray = arr[0] × No of subarrays contains arr[0] + arr[1] × No of subarrays contains arr[1]

+ arr[2] × No of subarrays contains arr[2] + . . . - - - . . . N-1 sum = 0

for (i = 0 ; i < N ; i++)

{

contri of i^{th} element = A[i] × sum of subarrays

$(i+1)(N-i)$

sum += contri of i^{th} element

}

return sum

$\boxed{\begin{array}{l} TC : O(N) \\ SC : O(1) \end{array}}$

TC: $O(N^3)$
SC: $O(1)$
$\xrightarrow{PF}$
TC: $O(N^2)$
SC: $O(N)$
$\xrightarrow{CF}$
TC: $O(N^2)$
SC: $O(1)$

$\downarrow$ Contri Tech

TC: $O(N)$
SC: $O(1)$

Sliding window [Nothing but carry Forward]

Q. Find No of subarrays having size = K (arr[N])

$N=7$
$K=4$

$a_0 \quad a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5 \quad a_6$

ans = 4

ending points

$a_0 \quad a_1 \quad a_2 \; \ldots \; a_{K-1} \quad a_K \quad a_{K+1} \; \ldots \; a_{N-1}$

$[K-1, N-1]$

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

$$b-a+1 = (N-1)-(K-1)+1$$
$$= N-1-K+1+1$$
$$= \boxed{N-K+1}$$

$k = 2$

$a_0 \quad a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5$

Q. Given an arr[N]. Print start & ending indices of each subarray of len K

N = 5 K = 3

[0, 1, 2, 3, 4]

[0, 2]
[1, 3]
[2, 4]

```
i = 0, j = K-1
while( j < N)
{
    print ( i + " - " + j)
    i++
    j++
```

Q. Given an arr[N]. Print maximum subarray sum of len = K

N = 10 K = 5

$$\begin{array}{c} [\,-3, & 4, & -2, & 5, & 3, & -2, & 8, & 2, & -1, & 4\,] \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{array}$$

s	e	sum
0	4	7
1	5	8
2	6	12
3	7	16
4	8	10
5	9	11

ans = 16

BF   For every subarray of len K. Iterate & get sum. maintain max

$i = 0,\ j = K-1$

```
while( j < N )
{
    ( i-j )

    sum = Calculate sum of i-j elements
    maxi = max(maxi, sum)

    i++
    j++
}
return maxi
```

TC: $O(N^2)$

SC: $O(1)$

A - 2 Use Prefix sum

// construct PF[]

i = 0, j = K-1

while(j < N)

{

(i - j)

sum = Calculate sum of i - j elements

use PF

maxi = max(maxi, sum)

i++

j++

}

return maxi

Cal PF

TC: O (N.1 + N) : O(N)

SC: O (N)

PF[]

A-3 Carry-Forward [Sliding window

[-3, 4, -2, 5, 3, -2, 8, 2, -1, 4]
0 1 2 3 4 5 6 7 8 9

[0 1 K-1, K]

```
sum = 0
for( 0 to K-1)          K ←
{
  sum += a[i]

maxi = sum

i = 1 , j = K
while(j < N)            N-K
{
  sum = sum - A[i-1] + A[j]
  maxi = max(maxi, sum)
  i++
  j++

return maxi
```

TC: O(N)
SC: O(1)

$$\begin{tikzcd}
\begin{array}{l} Tc: O(N^2) \\ Sc: O(1) \end{array} \arrow[r, "DF"] & \begin{array}{l} Tc: O(N) \\ Sc: O(N) \end{array} \arrow[d, "\text{CF (Sliding window)}"] \\
& \begin{array}{l} Tc: O(N) \\ Sc: O(1) \end{array}
\end{tikzcd}$$

2D matrices

-
-
-
-

Doubts