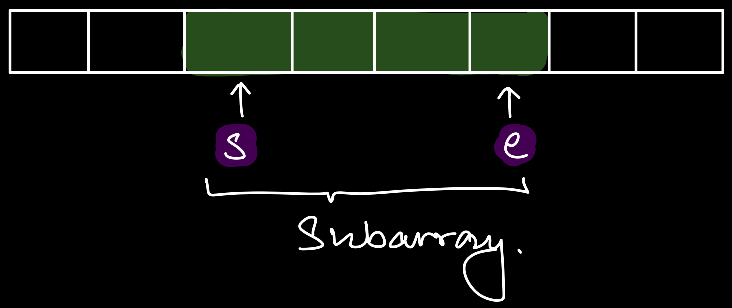
→ Prefix sum : Range sum queries

→ Carry forward

[s, e] ⇒ Subarray

SUBARRAY :- Contiguous part of the Array.



1) Complete Array is a subarray of itself.
2) Single element is also a subarray.

Note: We are only going to consider NON empty subarrays.

Ex

A : 3 4 5 6 -2 8 10

1) 5, 6, -2 ✓
2) 3, 4, 6, -2 ✗
3) 8 ✓
4) -2 ✓
5) 3 10 ✗
6) 10 3 ✗
7) 6 5 4

Ex [5, 3, 5]

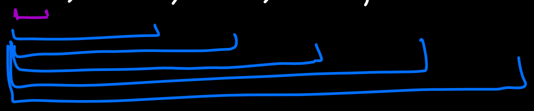
{5} ✓

{5, 3, 5} ✓

⇒ Subarray ⇒ [s, e] index

Quiz A: [4, 2, 10, 3, 12, -2, 15]

$$\begin{array}{ccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ 4, & 2, & 10, & 3, & 12, & -2, & 15 \end{array}$$



# of subarrays starting from index = 1

⇒ 6

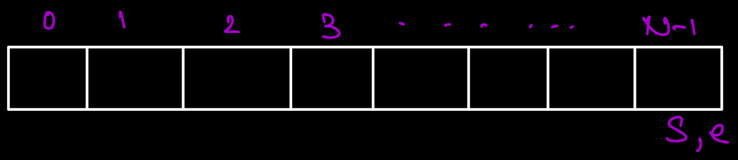
Quiz No. of subarrays in A:

$$\begin{array}{ccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ [4, & 2, & 10, & 3, & 12, & -2, & 15] \end{array}$$ N = 7

N = 7

$$\frac{7(7+1)}{2} = \underline{\underline{28}}$$

Quiz



Total no. of subarrays in an Array of size N.

= No. of subarrays starting at index = 0 ⇒ N

+ +

No. of subarrays starting at index = 1 ⇒ N-1

+ +

No. of subarrays starting at index = 2 ⇒ N-2

+ +

⋮

+ +

No. of subarrays starting at index = N-1 ⇒ 1

$\Rightarrow 1 + 2 + 3 + \dots + N-1 + N$

$\Rightarrow \frac{N(N+1)}{2} \Rightarrow \underline{\underline{O(N^2)}}$

Q. Print all the values of a subarray. s e

[s, e]

e-s+1

```
void printSubArray ( Arr, s, e ) {
    for ( i = s ; i <= e; i++ ) {
            print (Arr[i])
    }
}
```

TC : O(N)
(Worst Case)

SC : O(1)

Q. Find the sum of all the values of a subarray.

```
void   sumSubArray ( Arr, s, e) {
 sum = 0
 for( i = s; i <= e; i++) {
       sum += Arr[i];
 }
 print(sum);
```

TC: O(N)
{W.C}
SC: O(1)

3

Q. Print all the subarray of a given Array.

$A: \{ \overset{0}{2} \quad \overset{1}{8} \quad \overset{2}{9} \}$

s	e		
0	0	→	[2]
0	1	→	[2, 8]
0	2	→	[2, 8, 9]
1	1	→	[8]
1	2	→	[8, 9]
2	2	→	[9]

$A: [\overset{0}{1}, \overset{1}{2}, \overset{2}{3}, \overset{3}{4}]$

$(\overset{s}{0}, \overset{e}{0}) \quad (0,1) \quad (0,2) \quad (0,3)$

$(\underset{s}{1}, 1) \quad (1,2) \quad (1,3)$

$(\underset{s}{2}, 2) \quad (2,3)$

$(\underset{s}{3}, 3)$

$s \Rightarrow e \in [s, N-1]$

$e >= s$

```
O(N^2) { for ( s = 0 ; s < N ; s++ ) {
       {       for (e = s ; e < N ; e++) {
                       // s,e => SubArray.
         O(N) {        printSubArray(Arr, s, e);
               }
         }
```

$$TC: \underset{\text{Iterate over all subarray}}{\underset{\uparrow}{O(N^2)}} * \underset{\text{TC to print 1 subarray}}{\underset{\downarrow}{O(N)}} \Rightarrow \underline{\underline{O(N^3)}}$$

SC: $O(1)$

$$A: \{ \underset{\sqcup}{\overset{0}{2}} \quad \overset{1}{8} \quad \overset{2}{9} \}$$

$s = 0,\ e = 0 \Rightarrow 2$

$e = 1 \Rightarrow 2, 8$

$e = 2 \Rightarrow 2, 8, 9$

$s = 1,\ e = 1 \Rightarrow 8$

$e = 2 \Rightarrow 8, 9$

$s = 2,\ e = 2 \Rightarrow 9$

Q. Print the sum of every single subarray

$A : [\overset{0}{3} \quad \overset{1}{2} \quad \overset{2}{-1} \quad \overset{3}{4}]$

s	e		sum
0	0	[3] →	3
0	1	[3 2] →	5
0	2	[3 2 -1] →	4
0	3	[3 2 -1 4] →	8
1	1	[2] →	2
1	2	[2 -1] →	1
1	3	[2 -1 4] →	5
⋮	⋮	⋮	

Brute force

```
for ( s = 0 ; s < N ; s++ ) {
        for (e = s; e < N ; e ++) {
            // Sum of subarray from s to e
            sumSubArray ( Arr, s, e ); → O(N)
        }
}
```

TC : $O(N^3)$

SC : $O(1)$

Using PS

$$\boxed{Sum[s,e] = PS[e] - PS[s-1]}$$

// Build PS → O(N)

```
//        for ( s = 0 ; s < N ; s++ ) {
O(N²)         for (e = s; e < N ; e ++) {
                  // Sum of subarray from s to e
    O(1) [        sum = PS[e] - PS[s-1];   → O(1)
                  print(sum)
              }
          }
```

TC : $O(N^2)$

SC : $O(N)$ → Build PS

Can we do better ? YES

Q Print the sum of all subarrays starting at index = 2.

0	1	2	3	4	5	6
7	3	2	-1	6	8	2

s	e	
2	2	→ $a[2]$
2	3	→ $a[2] + a[3]$
2	4	→ $a[2] + a[3] + a[4]$
2	5	→ $a[2] + a[3] + a[4] + a[5]$
2	6	→ $a[2] + a[3] + a[4] + a[5] + a[6]$

```
Sum = 0
for ( i = 2 ; i < N; i++) {
        sum += A[i]
        print (sum);
}
```

0	1	2	3	4	5	6
7	3	2	-1	6	8	2
		s				↑ i

Sum = ~~0~~ ~~2~~ ~~1~~ ~~7~~ ~~15~~ 17

i = ~~2~~ ~~3~~ ~~4~~ ~~5~~ 6

2 , 1 , 7 , 15 , 17

#

```
for ( s = 0; s < N; s++) {
    sum = 0
    for ( i = s ; i < N; i++) {
        sum += A[i]
        print (sum);
    }
}
```

$\boxed{\begin{array}{l} TC : O(N^2) \\ SC : O(1) \end{array}}$ → Carry forward

$$A : \left[\begin{array}{cccc} 0 & 1 & 2 & 3 \\ 3 & 2 & -1 & 4 \end{array}\right]$$

$s = 0$, $sum = \cancel{0} + 3 + 2 - 1 + 4 = 8$

$i = \cancel{0}$
$\cancel{1}$
$\cancel{2}$
$\cancel{3}$
4

3, 5, 4, 8,
2, 1, 5

$s = 1$, $sum = \cancel{0}\ 2 - 1 + 4$

$i = \cancel{1}$
$\cancel{2}$
$\cancel{3}$
$\underline{\underline{4}}$

$s = 2$, $sum = 0$

$i = 2$

$\vdots$

Q Given an Array, find the sum of all subarray sums.

Google
FB.

A : { $\overset{0}{1}$, $\overset{1}{2}$, $\overset{2}{3}$ }

s	e		
0	0	[1]	→ 1
0	1	[1, 2]	→ 3
0	2	[1, 2, 3]	→ 6
1	1	[2]	→ 2
1	2	[2 3]	→ 5
2	2	[3]	→ 3

20

```
Sum = 0
for ( s = 0; s < N; s++) {
        for ( i = s ; i < N; i++) {
                sum += A[i]
        }
}
Print (sum);
```

TC : $O(N^2)$

SC : $O(1)$

$$A : \{\overset{0}{1}, \overset{1}{2}, \overset{2}{3}\}$$

s	e				
0	0	[1]	$\rightarrow$ 1	$\Rightarrow$	$a[0]$
0	1	[1,2]	$\rightarrow$ 3	$\Rightarrow$	$a[0]+a[1]$
0	2	[1, 2, 3]	$\rightarrow$ 6	$\Rightarrow$	$a[0]+a[1]+a[2]$
1	1	[2]	$\rightarrow$ 2	$\Rightarrow$	$a[1]$
1	2	[2 3]	$\rightarrow$ 5	$\Rightarrow$	$a[1]+a[2]$
2	2	[3]	$\rightarrow$ 3	$\Rightarrow$	$a[2]$
			20		$3 \times a[0] + 4 \times a[1] + 3 \times a[2]$

$$\Downarrow$$

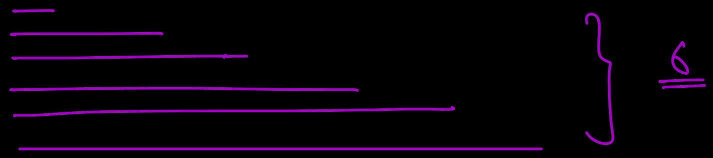
$$3 \cdot 1 + 4 \cdot 2 + 3 \cdot 3$$

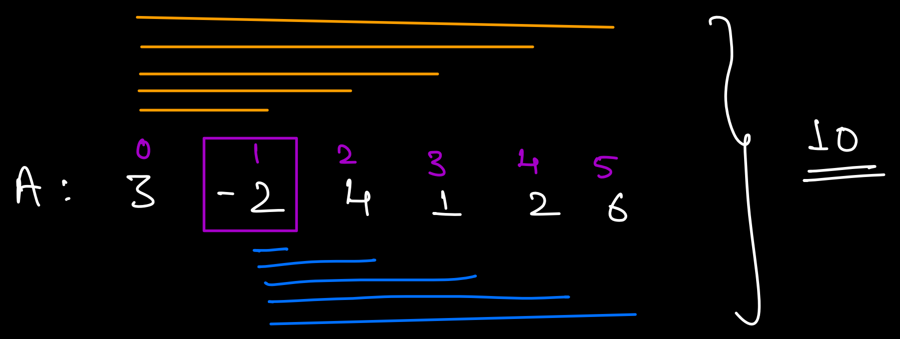
$$3 + 8 + 9 = \underline{\underline{20}}$$

\# In How many subarrays an element will be present.

$\Rightarrow$ No. of subarrays, index = 0 will be present

$$A : \overset{0}{3} \quad \overset{1}{-2} \quad \overset{2}{4} \quad \overset{3}{1} \quad \overset{4}{2} \quad \overset{5}{6}$$





$s \leq 1 \Rightarrow |s| = 2$
$e \geq 1 \Rightarrow |e| = 5$ } $\underline{\underline{10}}$

$$A: \begin{array}{cccccc} 0 & 1 & 2 & \boxed{3} & 4 & 5 \\ 3 & -2 & 4 & \boxed{1} & 2 & 6 \end{array}$$

* **Index = i** will be present in the subarrays

$\Rightarrow$ $s \leq i$

$\Rightarrow$ $s \in [0, i]$

$|s| = i+1$

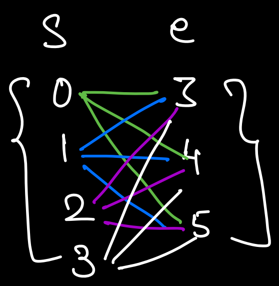
$\Rightarrow$ $e \geq i$

$e \in [i, N-1]$

$|e| = N - \cancel{1} - i + \cancel{1}$

$= \underline{\underline{N-i}}$

$\Rightarrow$ s e



$\Rightarrow$ $|s| \times |e|$

$\Rightarrow$ $4 \times 3 = \underline{\underline{12}}$

No. of subarrays, index = i will be present

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

Sum of all subarray sums

$$= \sum_{i=0}^{N-1} (i+1) \times (N-i) \times A[i]$$

```
sum = 0
for( i= 0; i < N; i++) {
        x = i+1
        y = N-i
        sum += x*y*a[i]
}
return sum;
```

TC : $O(N)$

SC : $O(1)$

Contribution Technique

* * *