Prefix Sum, Carry forward technique

Subarrays.

⇒ Contiguous part of an array (l, r)

starting index (l)

ending index (r)

$$r \geq l$$

⇒ Largest subarray $l = 0$
$r = N-1$

Complete array is also a subarray of itself

⇒ Single element can also be a subarray.

⇒ Empty array is also a subarray

⇒ But we will only talk about non empty subarray.

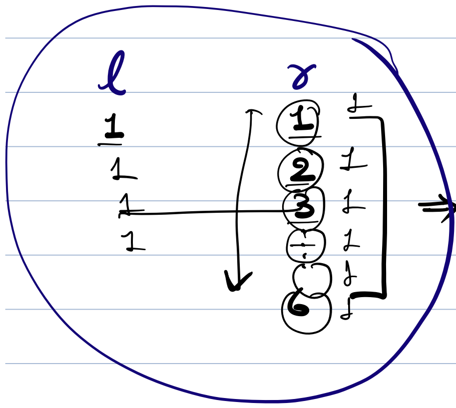
A: 3, 4, 5, 6, −2, 8, 10

5, 6, −2

3, 4, 6, −2

Quiz

	0	1	2	3	4	5	6
A:	4	2	10	3	12	−2	15



$\Rightarrow$ 6 subarrays

No. of subarrays in the given array.

$$A : \begin{array}{ccccccc} s & s & & & & & \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ 4, & 2, & 10, & 3, & 12, & -2, & 15 \\ e & e & e & e & e & e & e \\ & s & & & & & e \end{array} \Rightarrow 7 \text{ ele.}$$

Starts at	Ends at	
0	[0, 6]	$\Rightarrow$ 7

4, [4, 2], [4, 2, 10], [4, 2, 10, 3]

1	[1, 6]	$\Rightarrow$ 6
2	[2, 6]	$\Rightarrow$ 5
3	[3, 6]	$\Rightarrow$ 4
4	[4, 6]	$\Rightarrow$ 3
5	[5, 6]	$\Rightarrow$ 2
6	[6, 6]	$\Rightarrow$ 1
		28

$$\frac{7 \times 8}{2} = \frac{56}{2} = 28$$

Q. No. of non Empty subarrays in an array of size N

subarrays = # SA. starting at 0 = N

+ +

" " " 1 = N-1

+ +

" " " 2 = N-2

+

" " " 3

⋮ ⋮

subarrays starting at N-1 = 1

$$1 + 2 + 3 + 4 \dots N = \frac{N \times (N+1)}{2}$$

No. of subarrays $= \frac{N \times (N+1)}{2}$

Q. Print all values of a subarray.

(s, e)

Code

```
void printSubarray(A[], s, e) {
```

```
⇒ for (i = s; i <= e; i++) {
        print (A[i]);
  }
}
```

Q Find the sum of a given subarray.

Q Print all subarrays of a given array of size N.

	0	1	2
A :	2,	8,	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]

Code

```
si → for (i = 0; i < N; i++) {
ei → for (j = i; j < N; j++) {
```

$\Rightarrow O(n^2)$

// Subarray i to j

print the subarray

```
for (K = i ; K <= j ; K++) {
        print A[K];
}
```

}

}

$$T.C. = O(n^3)$$

$$\underbrace{O(N^2)}_{\#\text{ of subarray}} \times \underbrace{O(N)}_{\text{Print}}$$

i = 0

j = 0   j = 1   j = 2   j = 3 ⇒ N

print   print   print   print ⇒ O(N)

i = 1

j ⇒ N-1

i = 2

j ⇒ N-2

:

1

Q Print the sum of every single subarray.

A: 3, 2, -1, 4
0 1 2 3

s	e			
0	0	[3]	⇒	3
0	1	[3, 2]	⇒	5
0	2	[3, 2, -1]	⇒	4
0	3	[3, 2, -1, 4]	⇒	8

⋮

Code

$O(n^2)$

```
for (i = 0; i < N; i++) { // fix i
    for (j = i ; j < N; j++) {
        sum = 0;
        for (K = i ; K <= j ; K++) {
            sum += A[K];
        }
        print (sum);
```

$O(1)$ ⇐ prefix sum

}

}

$$T.C. = O(n^3) \xrightarrow{\text{PS array}} O(N^2) + O(N)$$

$$S.C. = O(N)$$

$$\#\ \text{subarray} = \frac{(N)(N+1)}{2} = \frac{n^2+n}{2}$$

$$O\left(\frac{n^2}{\cancel{2}} + \frac{\cancel{n}}{2}\right) = \underline{\underline{O(n^2)}}$$

1, 2, 3, 4, 5, 6 . . . . . . $\underline{\underline{n^2}}$ $\Rightarrow$ $O(n^2)$

Q Print sum of all subarrays starting from the index 3 (no extra space allowed)

	0	1	2	3	4	5	6
A:	7,	3,	2,	-1,	6,	8,	2

$$T.C. = O(n^2) \Rightarrow O(N)$$

i = 2, 3, 4, 5, 6,

S

	0	1	2	3	4	5	6
A:	7,	3,	2,	-1,	6,	8,	2
			e	e			

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]

Code

```
→ s = 2;   ⇒
  sum = 0;
  for (j = s ; j < N ; j++) {
        sum = sum + arr[j];
        print (sum);
  }
```

Code - 2

[i, i] ⇒ single element sum.

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

Google
FB.
Amazon

Q Given an array of size N.
find the sum of all subarray sum.

0 1 2
A : 1, 2, 3

S	e			Sum		
0	0	⇒		1	→	a[0]
0	1	⇒	(1)	3	→	a[0] + a[1]
0	2	⇒	(2)	6	→	a[0] + a[1] + a[2]
1	1	⇒	(3)	2	→	a[1]
1	2	⇒	(4)	5	→	a[1] + a[2]
2	2	⇒		3	⇒	a[2]

$\Sigma = 20$

$3 \times a[0] + 4 \times a[1] + 3 \times a[2]$

(1) Brute force

```
totalSum = 0;
for (i = 0; i < N; i++) {
    Sum = 0;
    for (j = i; j < N; j++) {
        Sum = Sum + A[j];
    }
    totalSum = totalSum + Sum;
}
```

}

print (totalSum) ;

T.C. = $O(N^2)$ rejected.

$\Downarrow$

$O(N)$ $\Rightarrow$ $O(1)$

$\Downarrow$

no. of elements present in the array.

Contribution Technique

$\Rightarrow$ If we are able to calculate the contribution of each element towards the sum.

Contribution = # times the element is occurring $\times$ the value of the element.

Total no. of subarrays, a given element is a part of

Quiz

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

s

$[0, 1]$

$\diagdown\ \diagup$

2

$s \le 1$

e

$e \ge 1$

⇒ # subarrays of which an element on index i will be a part of

$$\boxed{(s \le i)\ \&\&\ (e \ge i)}$$

$$\Downarrow \qquad\qquad\qquad \Downarrow$$

$[0 - i]$ $\qquad$ $[\overset{l}{i}, \overset{r}{N-1}]$ $\qquad$ $\Rightarrow (r - l + 1)$

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

$= (N-i)$

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

Total no. of subarrays $= |s| \times |e|$

$= \underline{(i+1) \times (N-i)} \Rightarrow$

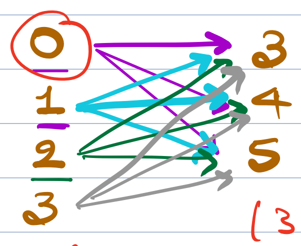
	0	1	2	3	4	5
A :	3,	−2,	4,	1,	2,	6

$i = 3$

$[0, 2]$

$s \le i$ && $e \ge i$

s, e



$[0 - 3]$

4

$[3, 5]$

$\Downarrow$

3

$[4, e]$

$s = 0$

$[0, 3]$
$[0, 4]$
$[0, 5]$

$s = 1$

$[1, 3]$
$[1, 4]$
$[1, 5]$

$s = 2$

$[2, 3]$
$[2, 4]$
$[2, 5]$

$s = 3$

$[3, 3]$
$[3, 4]$
$[3, 5]$

Code

```
Sum = 0;

for (i = 0;  i < N;  i++) {
        s = (i+1);
        e = (N-i);

        count = s x e;
        contribution = count x A[i];
```

sum = sum + contribution;

}

T.C. = O(N)

S.C. = O(1)

Doubts

A:
$$\begin{array}{ccc} 0 & 1 & 2 \\ 1, & 2, & 3 \end{array}$$

— [1] [2] [3]

— [1, 2] [2, 3]

— [1, 2, 3]

SS