Subarray

↳ continous part of an array

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

[9 3 7] → ✓

[9 3 8] → ✗

i) a single ele is also a subarray.

ii) a complete array is also a subarray.

iii) order matter $(s <= e)$

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

[4] → ✓

[4 9 3 7 8] → ✓

[8 7 3] → ✗

A particular subarray is defined using start and end

$s = 2$ subarray ⇒ [9 3 7 8]

$e = 5$

len ⇒ $e - s + 1$

$$A = \begin{array}{cccc} [\;3 & 4 & -2 & 5\;] \\ 0 & 1 & 2 & 3 \end{array}$$

$s = 0, \quad e = 0 \quad 1 \quad 2 \quad 3$

$0, 0 \rightarrow 3$

$0, 1 \rightarrow 3 \quad 4$

$0, 2 \rightarrow 3 \quad 4 \quad -2$

$0, 3 \rightarrow 3 \quad 4 \quad -2 \quad 5$

$s = 1, \quad e = 1 \quad 2 \quad 3$

$1, 1 \rightarrow 4$

$1, 2 \rightarrow 4 \quad -2$

$1, 3 \rightarrow 4 \quad -2 \quad 5$

$s = 2, \quad e = 2 \quad 3$

$2, 2 \rightarrow -2$

$2, 3 \rightarrow -2 \quad 5$

$s = 3, \quad e = 3$

$3, 3 \rightarrow 5$

$$A = [\,a_0 \quad a_1 \quad \cdots\cdots \quad a_{n-1}\,]$$

how many subarray starting from 0th index → n

how many subarray starting from 1st index → $n-1$

⋮

how many subarray starting from $(n-1)$th index → 1

$$n + (n-1) + (n-2) + \dots + 1$$

total subarray = $\frac{n(n+1)}{2}$

Q.1 Print a subarray of A[ ] from s to e.

```
void  print (int [ ]A, int s, int e) {

        for (int  k = s;  k <= e;  k++) {
                SOP (A [k]);
        }

}
```

Q.2 Print all subarrays of a given array.

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

s=0, e=0	3
s=0, e=1	3 -2
s=0, e=2	3 -2 4

s=1, e=1	-2
s=1, e=2	-2 4

s=2, e=2	4

```
void solve (int [] A) {
    int n= A.length;
    for (int s= 0; s < n; s++) {
        for (int e=s; e<n; e++) {
            // point subarray from s to e
            for(int K=s; K<=e; K++) {
                sop (A[K] + " ");
            }
            sopln ( );
        }
    }
}
```

A=	3	1	5
	0	1	2

s	e	K
0	0	0,0 → 3
	1	0,1 → 3 1
	2	0,2 → 3 1 5
1	1	1,1 → 1
	2	1,2 → 1 5
2	2	2,2 → 5

TC: $O(n^3)$

Q-3 Given an array print sum of every subarray of A[].

A = [3 -2 6]
      0   1 2

s e	subarray	sum
0, 0	[3]	3
0, 1	[3 -2]	1
0, 2	[3 -2 6]	7
1, 1	[-2]	-2
1, 2	[-2 6]	4
2, 2	[6]	6

i) TC: $O(N^3)$, SC: $O(1)$

ii) Improvisation → prefix sum

```
void solve (int [] A ) {

    int[] ps = prefixSum (A);
    int n = A.length;

    for (int s = 0; s < n; s++) {

        for(int e = s; e < n; e++) {

            // sum of subarray from s to e
            if (s = = 0) {
                SOPln(PS[e]);
            }
            else {
                SOPln (PS[e] - PS[S-1]);
            }
        }
    }
}
```

TC: $O(n^2)$

SC: $O(n)$

iii) further improvisation

$$A = \begin{array}{ccccccc} [& 3 & -2 & 4 & 5 & 1 &] \\ & 0 & 1 & 2 & 3 & 4 & \end{array}$$

s = 1   e : 1 2 3 4

(1, 1) $sum = A[1]$

(1, 2) $sum = \underbrace{A[1]}_{sum} + A[2]$

(1, 3) $sum = \underbrace{A[1] + A[2]}_{sum} + A[3]$

(1, 4) $sum = \underbrace{A[1] + A[2] + A[3]}_{sum} + A[4]$

✓

```
void  solve (int [ ] A ) {

    int n= A.length;

    for (int s = 0; s < n; s++) {
    |       int sum = 0;
    |       for (int e = s; e < n; e++) {
    |       |       sum += A [e];
    |       |       SOPln (sum);
    |       }
    }
}
```

TC: O(n²) , SC: O(1)

$$A = \begin{array}{ccccc} [& 3 & -2 & 4 &] \\ & 0 & 1 & 2 & \end{array}$$

s	e	sum	o/p
0	0	3	3
	1	3+(-2)	1
	2	3+(-2)+4	5
1	1	-2	-2
	2	-2 + 4	2
2	2	4	4

Q.4 Find totalsum of all subarrays sum. { google, Fb }

A = [3 -2 6]
0 1 2

s e	subarray	sum
0, 0	[3]	3
0, 1	[3 -2]	1
0, 2	[3 -2 6]	7
1, 1	[-2]	-2
1, 2	[-2 6]	4
2, 2	[6]	6
		19

```
void solve (int [] A ) {

    int n = A.length;
    int ts = 0;
    for (int s = 0; s < n; s++) {
            int sum = 0;
            for (int e = s; e < n; e++) {
                    sum += A [e];
                    ts += sum;
            }
    }
    return ts;
}
```

TC: O(n²) , SC: O(1)

Expected TC : O(n)

A = [3 -2 6]
      0   1   2

s e	subarray	sum
0, 0	[3]	3
0, 1	[3 -2]	1
0, 2	[3 -2 6]	7
1, 1	[-2]	-2
1, 2	[-2 6]	4
2, 2	[6]	6
		19

$A[0]$

$A[0] + A[1]$

$A[0] + A[1] + A[2]$

$A[1]$

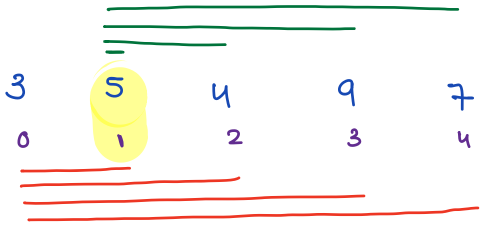
$A[1] + A[2]$

$A[2]$

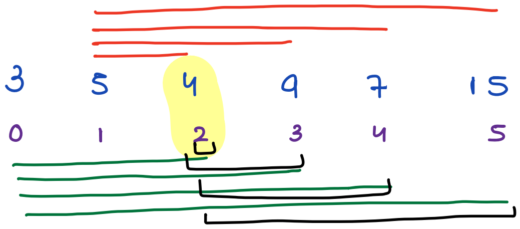
$3 * A[0] + 4 * A[1] + 3 * A[2]$

$3 * 3 + 4 * (-2) + 3 * 6 = 19$

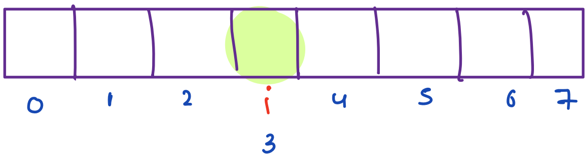
A[i] is coming in how many subarrays ?



Count = 8



count = 12



a to b

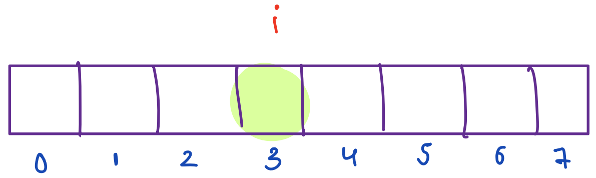
⇒ b-a+1

sp	ep
0	3
1	4
2	5
3	6
	7

possible start points ⇒ 0 to i : i+1 sp

possible end points ⇒ i to n-1 : n-i ep

count of valid subarray = (i+1) * (n-i)



a to b

⇒ b-a+1

sp	ep
0	3
1	4
2	5
3	6
	7

possible start pt : 0 to i ⇒ i+1 sp

possible end pt : i to n-1 ⇒ n-i ep

count of subarrays = $(i+1)*(n-i)$

```
int    solve (int [] A) {

    int ans= 0;

    int n= A.length;

    for (int i=0; i<n; i++) {

        ans += A[i] * ((i+1)*(n-i));

    }

    return ans;

}
```

$$A = \begin{array}{cccc} [\,3 & -2 & 6\,] \\ 0 & 1 & 2 \end{array}$$

i	
0	$3*(1*3)=9$
	+
1	$-2*(2*2)=-8$
	+
2	$6*(3*1)=18$
	19

TC: O(n)

Doubts

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

sum(1, 2) => ps[2] - ps[0]

$\overline{A[\cancel{0}] + A[1] + A[2]} - A[\cancel{0}]$

=> A[1]+A[2]

1 - (-2) = 3

```
for(int  i=0; i<n; i++) {
      for (int j= i-1; j>=0; j++ ) {
               SOP();
      }
}
```

i	j	idx
0	-1	0
1	~~0~~ ~~1~~ ~~2~~ ~~3~~4.....	infinite

Array product puzzle

$$A = \begin{array}{ccccc} [\;1 & 4 & 2 & 3 & 5\;] \\ 0 & 1 & 2 & 3 & 4 \end{array}$$

$$ans = \begin{array}{ccccc} [\;120 & 30 & 60 & 40 & 24\;] \\ 0 & 1 & 2 & 3 & 4 \end{array}$$

ans[i] → product of all elements of A[ ], except

i^{th} element.

$$A = \begin{array}{ccccc} [\;1 & 4 & 2 & 3 & 5\;] \\ 0 & 1 & 2 & 3 & 4 \end{array} \qquad prod = 1*4*2*3*5$$

$$ans[i] = \frac{prod}{A[i]} \quad \Big\} \quad \text{manage edge cases (val} \to 0)$$

Note : don't use division.

$$\begin{array}{lllllll} A = & [\ 1 & 4 & 2 & 3 & 5\ ] \\ & 0 & 1 & 2 & 3 & 4 \end{array}$$

$ans[0] = sm[1]$

$ans[n-1] = pm[n-2]$

$pm = [\ 1 \quad 4 \quad 8 \quad 24 \quad 120\ ]$

$sm = [\ 120 \quad 120 \quad 30 \quad 15 \quad 5\ ]$

$ans = [\ 120 \quad 30 \quad 60 \quad 40 \quad 24\ ]$

$$ans[i] = \underbrace{mult(0, i-1)}_{pm[i-1]} \times \underbrace{mult(i+1, n-1)}_{sm[i+1]}$$

$pm[i]$ → multiplication of values from 0 to i

$pm[i] = pm[i-1] * A[i]$

$sm[i]$ → multiplication of values from i to n-1

$sm[i] = sm[i+1] * A[i]$