Subarrays →

→ Continuous portion of the array!

→ Single element ✓

→ whole array ✓

→ empty part ✗

	0	1	2	3	4	5	6	7	8
A :	2	4	6	2	9	3	2	5	7

indices : [0, 1, 2, 3] ✓

— : [1, 2, 4, 7] ✗

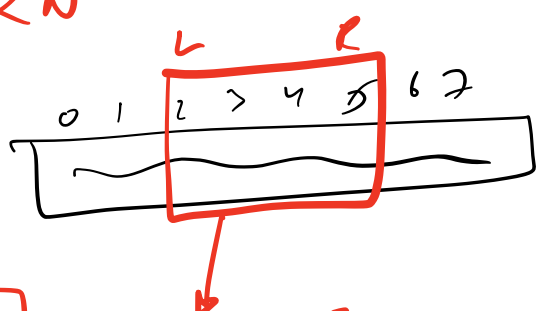
— : [4] ✓

④ Uniquely identify a S.A using [st, end] or [L, R]

$0 <= st <= end < N$

or

$0 <= L <= R < N$

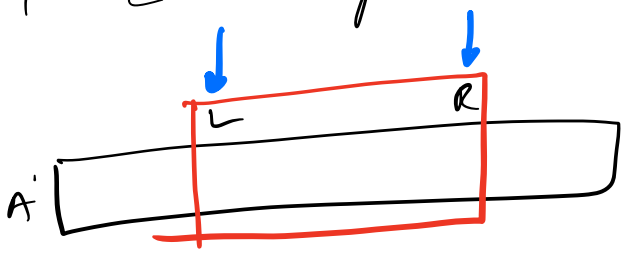


[2, 5]

elements in SA → R − L + 1

Q Given an array. Given [L & R], print the SA[L,R].

```
for(i=L; i<=R; i++){
    print(A[i]);
}
```



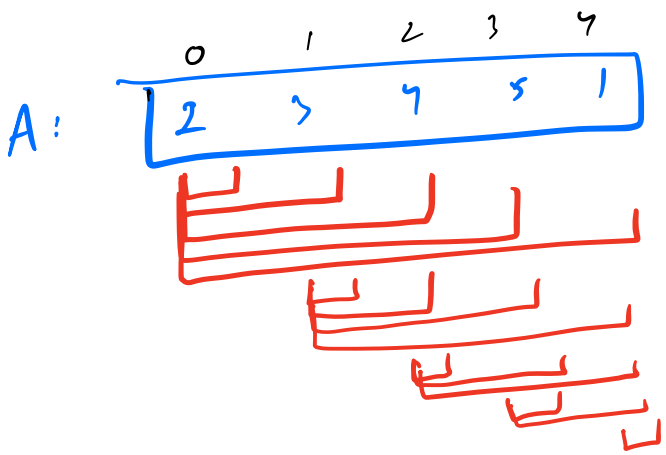
TC: O(R-L)

~~TC: O(N)~~

Q Given an array of size N.

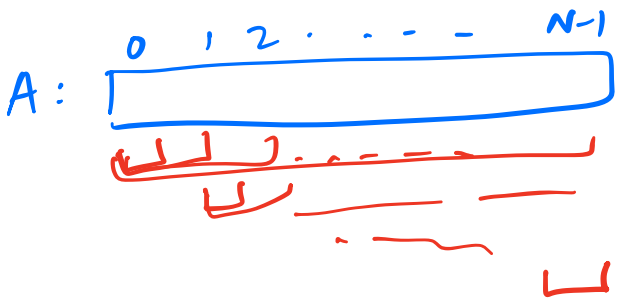
Find # SA.

N = 5



5 + 4 + 3 + 2 + 1 = 15

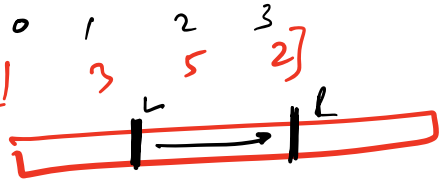
Generalize:



N + N-1 + . . . + 1

$$\#\ \text{SubArray} = \frac{N(N+1)}{2} \rightarrow O(N^2)$$

Q Given an array, print all the Sub-arrays!



[L, R]	→	S.A
[0, 0]	→	1
[0, 1]	→	1, 3
[0, 2]	→	1, 3, 5
[0, 3]	→	1, 3, 5, 2
[1, 1]	→	3
[1, 2]	→	3, 5
[1, 3]	→	3, 5, 2
[2, 2]	→	5
[2, 3]	→	5, 2
[3, 3]	→	2

```
for( L = 0; L < N; L++){
    for( R = L; R < N; R++){
        // [L, R]
        for( i = L; i <= R; i++){
            print(A[i]);
        }
        print( New Line);
    }
}
```

$$\frac{N(N+1)}{2} \times N$$

TC: $O(N^3)$

Q Given an array. Print all the Subarray sums.

A : [1 3 5 2] (indices 0 1 2 3)

[L, R]	S.A	Sum
[0, 0]	1	1
[0, 1]	1, 3	4
[0, 2]	1, 3, 5	9
[0, 3]	1, 3, 5, 2	11
[1, 1]	3	3
[1, 2]	3, 5	8
[1, 3]	3, 5, 2	10
[2, 2]	5	5
[2, 3]	5, 2	7
[3, 3]	2	2

```
for ( L = 0; L < N; L++) {
    for ( R = L; R < N; R++) {
        // [L, R] sum = 0
        for ( i = L; i <= R; i++) {
            sum += A[i];
        }
        print ( sum );
        print ( New Line);
    }
}
```

TC = O(N^3)

II) Use PS →

TC	SC
O(N)	O(N)

1. Build the PS Arr!

2.
```
for( L=0; L<N; L++) {
    for( R=L; R<N; R++) {
        // [L,R]
        if( L==0){
            print(PS[R]);
        }
        else {
            print( PS[R] - PS[L-1]);
        }
        print( newLine);
    }
}
```

TC: O(N²)

SC: O(1)

$TC = O(N + N^2)$

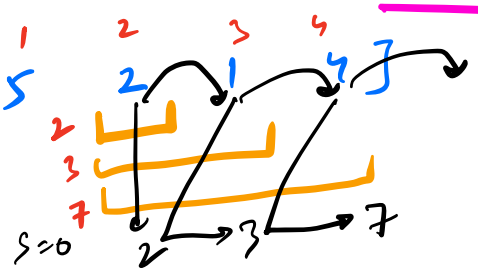
$TC = O(N^2)$

$SC: O(N)$ PS[]

Print all the S.A. sums starti at index 2!

Carry forward!

A : [3 5 2 1 4 7]




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

(*) Solving the prev problem

```
f(L = 0 ; L < N ; L++) {
    Sum = 0
    f(R = L ; R<N ; R++) {
        Sum += A[R];
        print(sum);
    }
}
```

TC = O(N²)

SC = O(1)

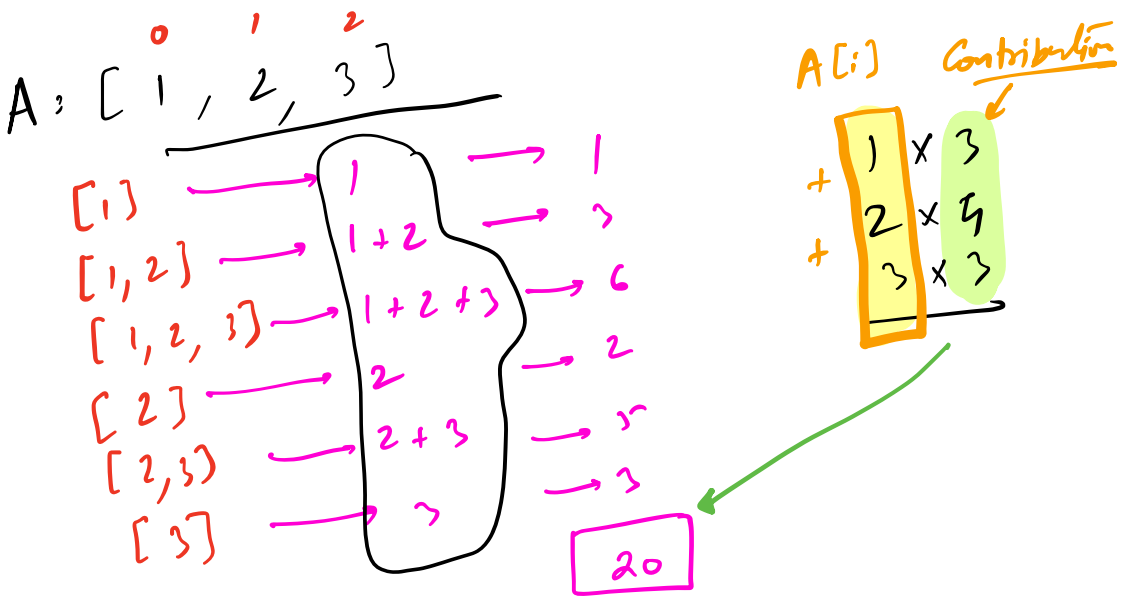
Q Given an array.
Find the sum of all the sub-array sums!

I

```
ANS = 0
for (L = 0; L < N; L++) {
    Sum = 0
    for (R = L; R < N; R++) {
        Sum += A[R];
        ANS += Sum
    }
}
// print ANS
```

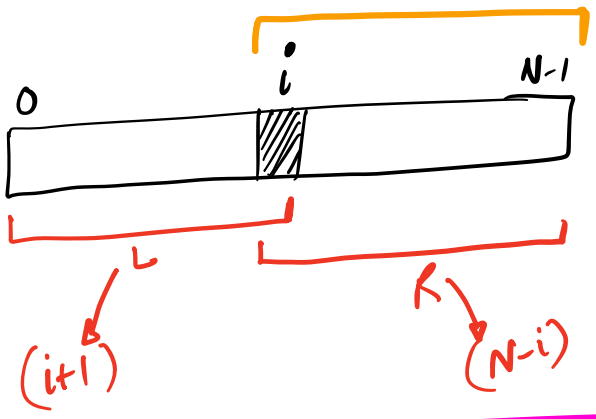
TC = O(N²)

SC = O(1)

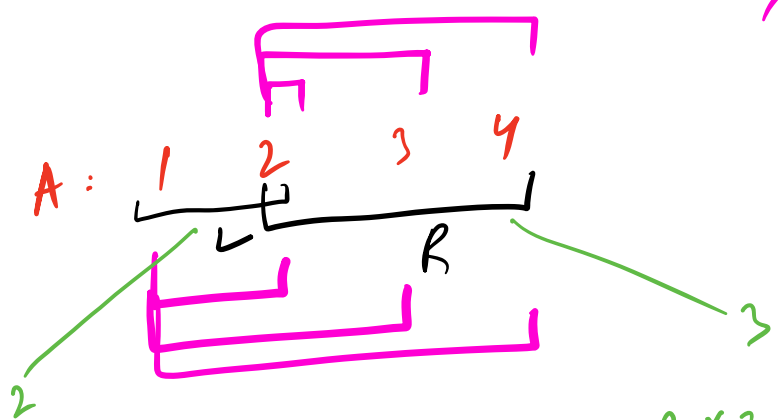


Idea: find contribution of every element!

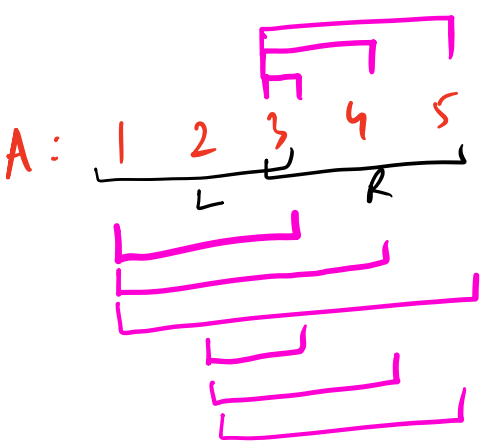
→ find the # SAs ith element is a part of!



SAs containing ith element = $(i+1)(N-i)$



$2 \times 3 = 6$!



$3 \times 3 = 9$!

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

```
ANS = 0
for( i = 0; i < N; i++) {
    ANS += A[i] x (i+1) (N-i);
}
print(ANS)
```

TC : O(N)

SC : O(1)