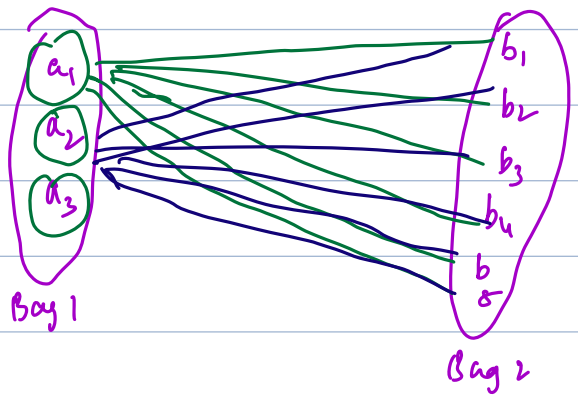
Question: Bag Problem

If there are 2 bags with 3 and 5 items in each bag, find no, of ways we can select one item from Bag1 and other from Bag2?

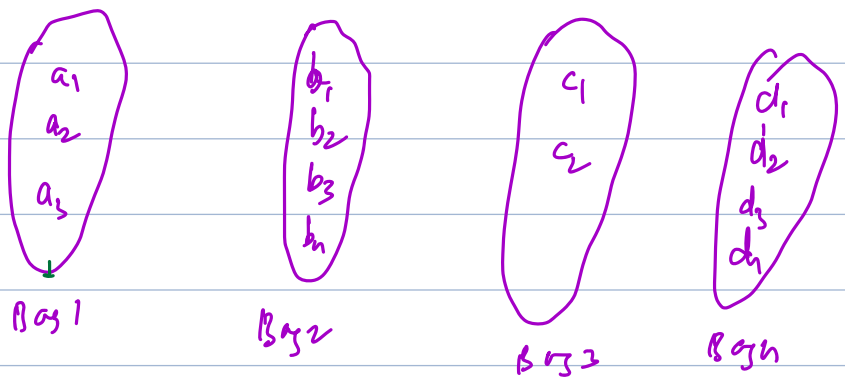


5 ⇒ (a_1, b_1) (a_1, b_2) (a_1, b_3) (a_1, b_4) (a_1, b_5)

5 ⇒ (a_2, b_1) (a_2, b_2) (a_2, b_3) (a_2, b_4) (a_2, b_5)

(a_3, b_1) (a_3, b_2) (a_3, b_3) (a_3, b_4) (a_3, b_5)

$5 + 5 + 5 + \ldots \Rightarrow 5 \times 3 = 15$



Ans :

$3 \times 4 \times 2 \times 4 = 96$ ways

Question: Find total sum of all subarrays

$N \leq 10^6$

A = [2, 8, 9, 4]

S	E	Subarrays	Sum
0	0	[2]	+ 2
0	1	[2, 8]	+ 10
0	2	[2, 8, 9]	+ 19
0	3	[2, 8, 9, 4]	+ 23
1	1	[8]	+ 8
1	2	[8, 9]	+ 17
1	3	[8, 9, 4]	+ 21
2	2	[9]	+ 9
2	3	[9, 4]	+ 13
3	3	[4]	+ 4
			106

Approach1:

```
long totalSum = 0;
for(start = 0; start < N; start++){
    long sum = 0;
    for(end = start; end < N; end++){
        sum += A[end];
        totalSum += sum;
    }
}
return totalSum;
```

T.C: $O(N^2)$

S.C: $O(1)$

$N \leq 10^6$

#subarrays: $\frac{N(N+1)}{2} \Rightarrow O(N^2)$ $\quad N^2 = (10^6)^2 = 10^{12}$

Approach:

If we iterate over all subarrays, we can never do less than $O(N^2)$

$A = [1 \; 2 \; 3]$ (indices 0 1 2) $N = 3$ #subarray $= \frac{3 \times 4}{2} = 6$

S	E	Sum	
0	0	A[0] ✓	=> 1
0	1	A[0] + A[1] ✓	=> 3
0	2	A[0] + A[1] + A[2] ✓	=> 6
1	1	A[1]	=> 2
1	2	A[1] + A[2]	=> 5
2	2	A[2]	=> 3

$= 20$

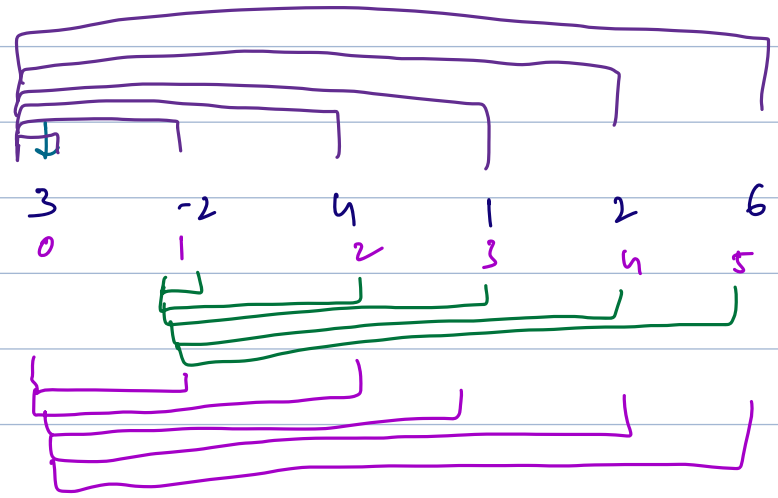
$3 \cdot A[0] + 4 \cdot A[1] + 3 \cdot A[2]$

$= 3(1) + 4(2) + 3(3) = 20$

what are these elements 3, 4, 3 ?

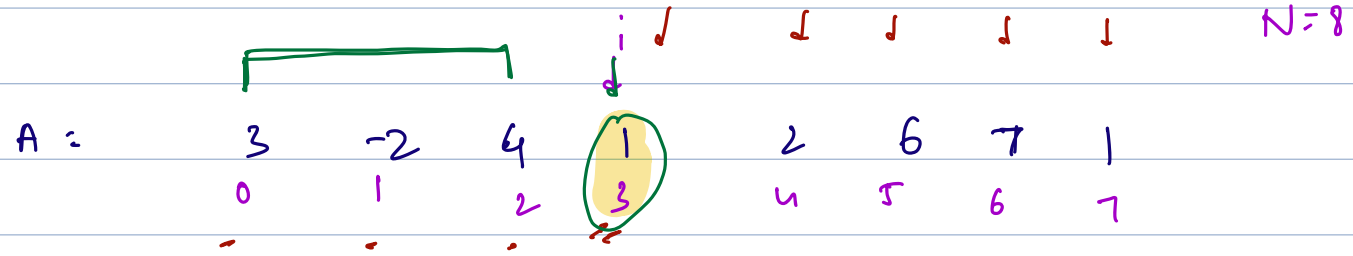
It denotes no. of subarrays in which A[i] is present

Task: For each element A[i], find no. of subarrays in which it is a part of.



#subarrays in which A[0] exist : 6 (A[0] x 6)

#subarrays in which A[i] exist : 5+5 = 10 (A[i] x 10)



Question: No. of subarrays in which A[3] is a part of

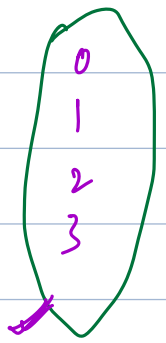
Can A[3] be a part of any subarray which starts after index 3 ? NO!

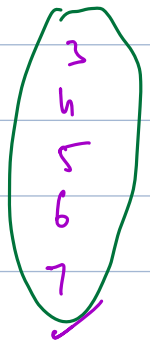
Start $\leq$ i => [0, i]

Will A[3] be a part of every subarray which starts before index 3 ? NO!

end $\geq$ i => end = [i, N-1]

i = 3





$= 4 \times 5 = \boxed{20}$

i ⇒ start: [0, i] ⇒ i+1

end: [i, N-1] ⇒ N-1-i+1 = N-i

#Subarrays which have A[i] : (i+1) x (N-i)

N = 6

	i	j				
A =	3	-2	4	-1	2	6
	0	1	2	3	4	5
(i+1) left	1	2	3	4	5	6
(N-i) right	6	5	4	3	2	1
#subarrays	6	10	12	12	10	6

(6 x 3) + (10 x -2) + (12 x 4) + (12 x -1) + (10 x 2) + (6 x 1) =

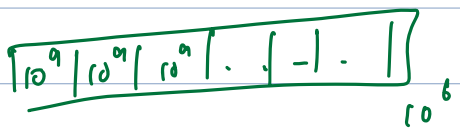
```
long sumSubarrays(int[] A, int N){
    long totalSum = 0;
    for(i=0; i<N; i++){
        totalSum += (long)A[i] * (i+1) * (N-i);
    }
    return totalSum;
}
```

$N \le 10^6$

$A[i] \le 10^9$

TC: O(N)

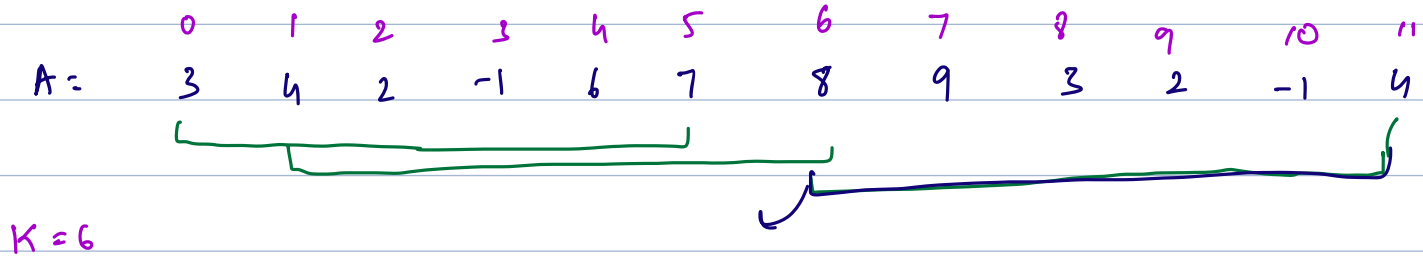
SC: O(1)



$10^6 \times 10^9 = 10^{15}$ → long value of any

Contribution Technique

Question: Given an array A of size N, print start and end indices of all subarrays of length K



K = 6

K = 5	K = 6	K
[0, 4]	[0, 5]	[0, K-1]
[1, 5]	[1, 6]	[1, K]
[2, 6]	[2, 7]	[2, K+1]
⋮	[3, 8]	⋮
	[4, 9]	
	[5, 10]	
[x, 11] (N-1)	[6, 11] (N-1)	[x, N-1]
11 - x + 1 = 5	Start: {0, 1, 2, 3, 4, 5, 6}	N-1 - x + 1 = K
x = 12 - 5 = 7	[0, 6] = 7	x = N - K
[7, 11]		[N-K, N-1]

Start : {0, 1, 2 . . . . . N-K}

[0, N-K] ⇒ N - K - 0 + 1 = N - K + 1

[a, b] : b - a + 1

N - K - 0 + 1 = N - K + 1

★★★★★

1st Subarray of size K : [0, K-1]

last subarray of size K : [N-K, N-1]

No. of subarrays of size K : N-K+1

```
printSubarrayK ( int [] A , int N, int K ){
    s = 0 ;
    e = K-1 ;
    while( s ≤ N-K ){                    // e ≤ N-1
        print( s, e );
        s++;
        e++;
    }
}
```

Question: Given an array, find the maximum subarray sum of length K

N = 10

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

K = 5

S	E	Sum
0	4	7
1	5	8
2	6	12
3	7	16
4	8	10
5	9	11