Sliding Window & Contribution technique

Q.1) Given an array of integer, find the total sum of all possible subarrays.

0	1	2

A[] = { 3, 2, 5 }

			sum
0	0	→ {3}	→ 3
0	1	→ {3, 2}	→ 5
0	2	→ {3, 2, 5}	→ 10
1	1	→ {2}	→ 2
1	2	→ {2, 5}	→ 7
2	2	→ {5}	→ 5

32 → Ans

Bruteforce: For all subarrays, calculate sum and add it in the answer

```
ans = 0
For L=0 to N-1 {
    for R=L to N-1 {
        sum = 0
        for(i=L to R) {
            sum = sum + A[i];
        }
        ans = ans + sum;
    }
}
return ans;
```

subarray sum L->R

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

* **Optimization:**

length of array.

```
int PF[N];
PF[0] = A[0];

for (i=1 to N-1)
{
    PF[i] = PF[i-1] + A[i];
}
```

TC: $O(N)$

```
ans=0
For L=0 to N-1 {
    for R=L to N-1 {
        if(L==0)  ans+= PF[R]
        else  ans+= PF[R]-PF[L-1]
    }
}
return ans;
```

TC: $O(N^2)$

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

SC: $O(N)$

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

0	0	$L=0$	$A[0] \rightarrow S$
0	1		$A[0]+A[1] \rightarrow S1$
0	2		$A[0]+A[1]+A[2]$
1	1	$L=1$	$A[1]$
1	2		$A[1]+A[2]$
2	2	$L=2$	$A[2]$

<u>Carry forward</u>

ans=0

for $L=0 \rightarrow N-1$

{

sum=0

TC: $O(N^2)$

SC: $O(1)$

for R = L to N-1

{

sum = sum + A[R]

ans = ans + sum;

}

}

return ans.

ans = ~~0~~ ~~3~~ ~~8~~ ~~18~~ ~~20~~ ~~27~~ 32

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

L	sum	R
0	~~0~~	~~0~~
	~~3~~	~~1~~
	~~5~~	~~2~~
	10	3
1	~~0~~	~~1~~
	~~2~~	~~2~~
	7	3

2   ~~0~~ 5   ~~2~~ 3

* If one element is used multiple times to calculate the answer ⇒ Contribution technique.

$$A[\,] = \{\overset{0}{3},\ \overset{1}{2},\ \overset{2}{5}\}$$

3 → 3
3 2 → 5
3 2 5 → 10
2 → 2
2 5 → 7
5 → 5

32

$= 3*3 + 2*4 + 5*3$
$= 9 + 8 + 15$

= 32

Contribution of A[i]

→ A[i] * (In how many subarray A[i] is present).

occurance of A[i]

i

Quiz:

0	1	2	3	4	5

A: [3, -2, 4, -1, 2, 6]

L	R	L	R
0	1	1	1
0	2	1	2
0	3	1	3
0	4	1	4
0	5	1	5

Ans = 10

i

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

L	R	L	R	L	R
0	2	1	2	2	2
0	3	1	3	2	3
0	4	1	4	2	4
0	5	1	5	2	5

Ans = 12

* Subarray with A[i]

\# starting index(L) = [0 i]

$= i - 0 + 1 = i + 1$

\# ending index(R) = [i N-1]

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

$= n-i$

Subarray having $A[i] = (i+1) * (n-i)$

$Ans = \sum_{i} A[i] * (i+1) * (n-i)$

```
ans = 0
for (i=0 to n-1)
{
    ans += A[i]*(i+1)*(n-i)
}
return ans;
```

$A = \{3, 2, 5\}$ (indices 0, 1, 2)

TC: $O(n)$
SC: $O(1)$

i	A[i]	(i+1)	(n-i)
0	3	1	3

$= 3*1*3$
$= 9$

1 2 2 2 $= 2*2*2$

$= 8$

2 5 3 1 $= 5*3*1$

$= 15$

$= \underline{\underline{32}}$

Break : 10.13 PM

Q. Find the total no. of subarray with length = K.

	0	1	2	3	4	5
A =	3	-2	4	-1	2	6

N = 6

K = 3 → Ans = 4

K = 5 → Ans = 2

1^{st} subarray → 0 ______ (K-1)

Last subarray → (N-K) ______ (N-1)

[st N-1] = K

$N-\cancel{1}-st+\cancel{1} = K$

N - st = K

st = N - K

Quiz!

0 1 2 3 4 5 6

$N = 7 \quad K = 4 \qquad [a_1, a_2, a_3, a_4, a_5, a_6, a_7]$

$= N - K + 1 = 7 - 4 + 1 = 4$

$[0 \rightarrow N-K]$

9.) print all start and end indices of length K.

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

$N = 6 \quad K = 3$

L	R	
0	2	$(0+3-1)$
1	3	$(1+3-1)$
2	4	$(2+3-1)$
3	5	$(3+3-1)$

```
for L->0 to(N-K)
{
    R=L+K-1
    print(L, R)
}
```

TC: O(N)

SC: O(1)

* Given an integer array, find max subarray sum for all subarreys with length = K   N = 6, K = 3

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

5   1   5   7

Ans = 7

L	R
0	2
1	3
2	4
3	5

[L R]

$R - L + 1 = K$

$R = K - 1 + L$

Bruteforce:

```
ans = INT_MIN

for L -> 0 to N-K
{
    R = L + K - 1;
    sum = 0;
    for (i = L to R)
    {
        sum = sum + A[i];
    }
    if (sum > ans)
    {
        ans = sum;
    }
}
```

TC = O(K)

$$TC = O((N-K+1) * K)$$

$$K = 1 \quad = (N-1+1) * 1 \quad = N$$

$$K = N \quad = (\cancel{N} - \cancel{N} + 1) * N \quad = N$$

$$K = \underline{N} \quad = \left(N - \underline{N} + 1\right) * \underline{N}$$

$$= \left(\frac{N}{2}+1\right) * \frac{N}{2}$$

$$= \frac{N^2}{4} + \cancel{\frac{N}{2}}$$

$$= N^2 * \cancel{\frac{1}{4}}$$

$$= N^2 \rightarrow \boxed{O(N^2)}$$

Prefix sum

SC: O(N) {

PF[N];

PF[0] = A[0]

TC: O(N) {

for(i=1 to N-1)

```
    PF[i] = PF[i-1] + A[i];
}

ans = INT_MIN;

for(L=0 to N-K)
{
    R = L+K-1;
    if(L==0)
    {
        sum = PF[R];
    }
    else
    {
        sum = PF[R] - PF[L-1];
    }
    if(sum > ans)
    {
        ans = sum;
    }
}

return ans;
```

TC: $O(N+(N-K+1)) \rightarrow O(N)$

SC: $O(N)$

$N=6,\ K=3$

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

5

$= 5 + A[3] - A[0]$
$= 5 + (-1) - 3$
$= 5 - 4$
$= 1$

$= 1 + A[4] - A[1]$
$= 1 + 2 - (-2)$
$= 1 + 2 + 2$
$= 5$

Sliding Window: fixed length subarray

L	R	Sum
0	2	$A[0]+A[1]+A[2] = 5$
1	3	$sum - A[0] + A[3] = 1$
2	4	$sum - A[1] + A[4] = 5$
3	5	$sum - A[2] + A[5] = 7$

Ans = 7.

```
sum=0
for(i=0 to K-1)
{
    sum = sum + A[i];
}

ans= sum;
for( L=1 to N-K)
{
    R = L+K-1;
    sum = sum - A[L-1]+A[R]
    if(sum > ans)
    {
        ans= sum;
    }
}
return ans;
```

TC: O(N)

SC: O(1)

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

sum = ~~5~~ ~~1~~ ~~5~~ 7 ans = ~~5~~ 7

L	R	
1	3	newsum $= 5 - A[0] + A[3]$ $= 5 - 3 + (-1)$ $= 1$
2	4	$= 1 - A[1] + A[4]$ $= 1 - (-2) + 2$ $= 5$
3	5	$= 5 - A[2] + A[5]$ $= 5 - 4 + 6$ $= 7$

Ans = 7

*) Subarray is contiguous part of an array.

*) array length = n, how many subarray = $\frac{n*(n+1)}{}$

Doubt session

'''

'''

$(n-k+1) * k$

$k = \frac{n}{2}$

$K = n$

$= (\cancel{n} - \cancel{n} + 1) * n$

$= n * 1$

$= \underline{\underline{n}}$