Problem Solving - 2

Q.1 Equilibrium Index :-

Given an array 'A' of size 'N', find the Eq. Index.

if (sum(0 --- i-1) = sum(i+1, n-1))
return 'i'

#1 Brute force approach :-

for every index 'i' :

$O(N)$ ← iterate on left part → sum_L
$O(N)$ ← iterate on right part → sum_R

if (sum_L == sum_R) : return i;

T.C: $O(N^2)$
S.C: $O(1)$

#2: Prefix Sum Array

$sum_R = PS[n-1] - PS[i]$

A :

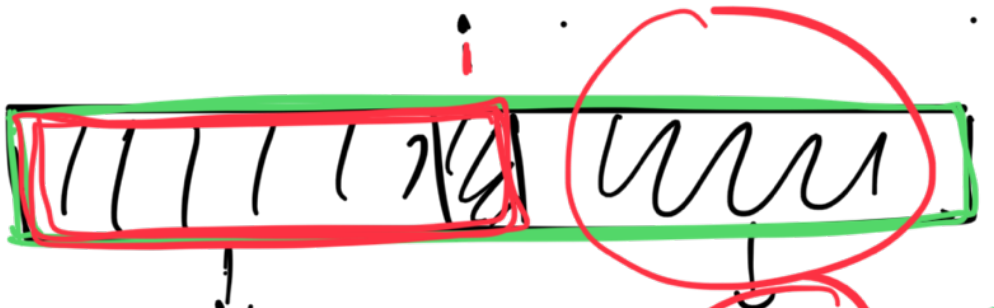
sum(0 --- i-1) = sum(i+1 --- N-1)

sum_L = PrefixSum[i-1]
sum_R = PrefixSum[n-1] - PrefixSum[i]

if (PS[i-1] == PS[n-1] - PS[i]) :

→ if (PS [i]

'i' is an Eq. Index.



$Sum_L = PS[i-1]$

$Sum_R = PS[n-1] - PS[i]$

```
for (i = 1 to n-1) {
    if (PS[i-1] == PS[n-1] - PS[i])
        return i;
}
```

PS[i] : sum(0, --- i)

sum(i+1, --- n-1) : PS[n-1] - PS[i]

	0	1	2	3	4	5
A :	-7	1	5	2	-4	3
PS :	-7	-6	-1	1	-3	0
		-7+1	-7+1+5			

$Sum_L = -7 + 1 + 5 = -1$

$Sum_R = -4 + 3 = -1$

'3' is an Eq. Index

PS[5] = PS[5] = 0

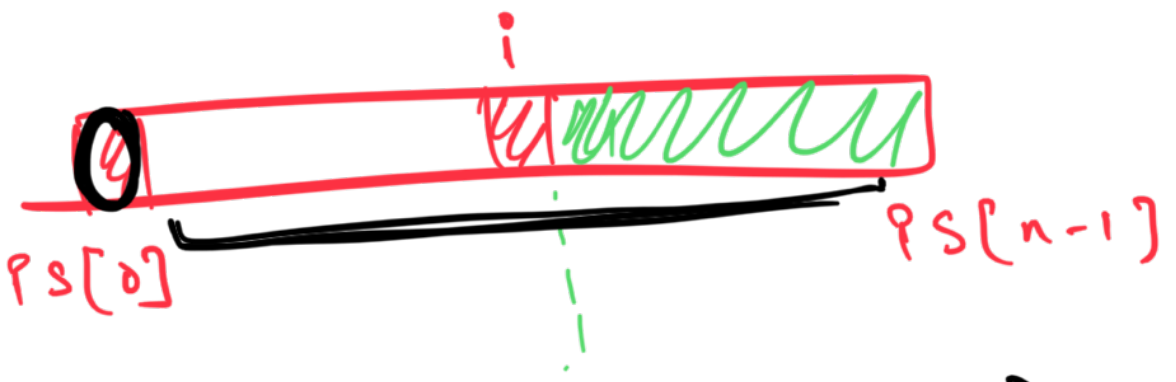
$PS[n-1] = PS[5] = 0$

$sum_L = PS[2] = sum(0,1,2)$

$sum_R = a[4] + a[5]$

$PS[n-1] = \underbrace{a[0] + a[1] + a[2] + a[3]} + a[4] + a[5]$

$= PS[5] - PS[3]$




```
# if (PS[n-1] - PS[0] == 0) return 0;

for (i = 1; i < n; i++) {
    if (sum_L == sum_R)
        return i;
}
```

A : { 5, 7, -1, -2, -4 } ⟹ '0'

(0 1 2 3 4)

$\boxed{sum_L = 0}$

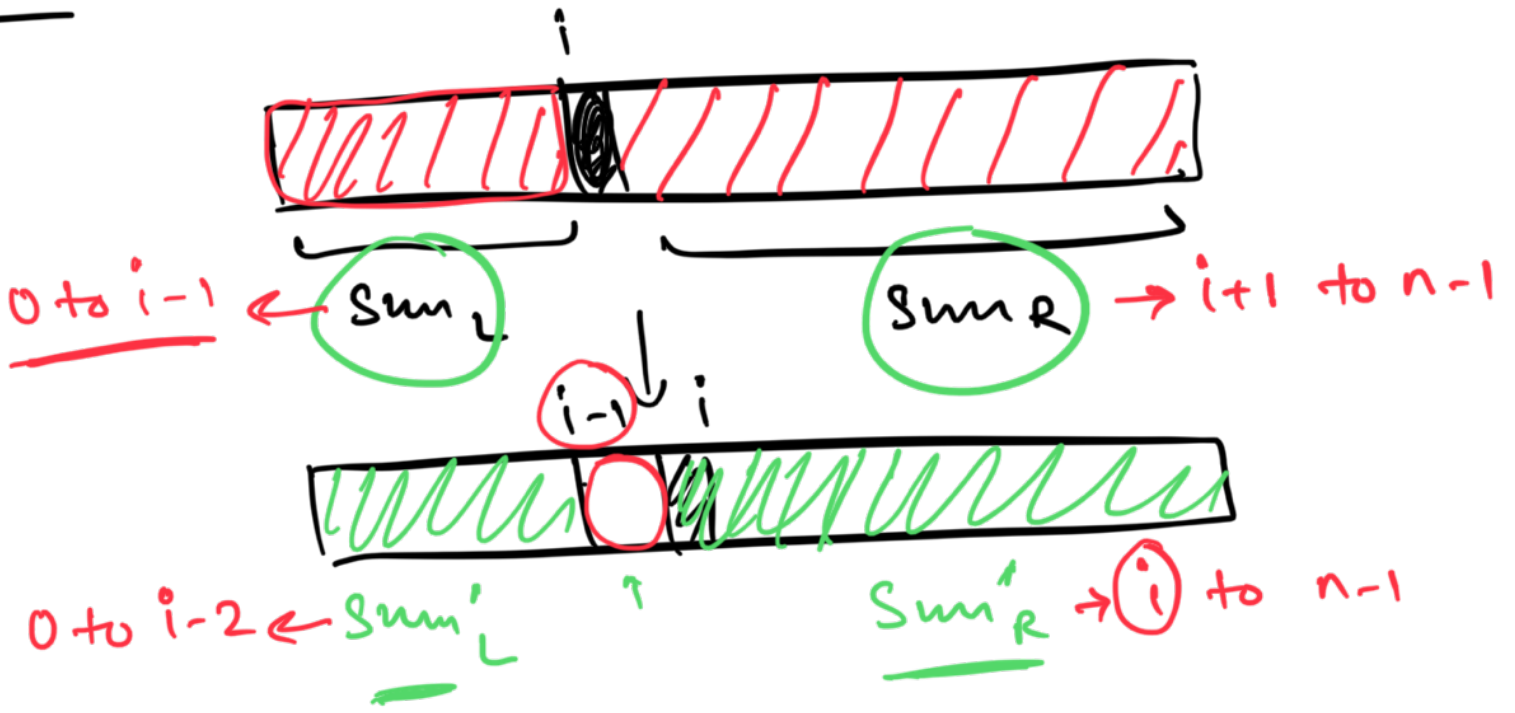
$sum_R = 7 + -1 - 2 - 4 = 0$

'0' is Eqbm Index.

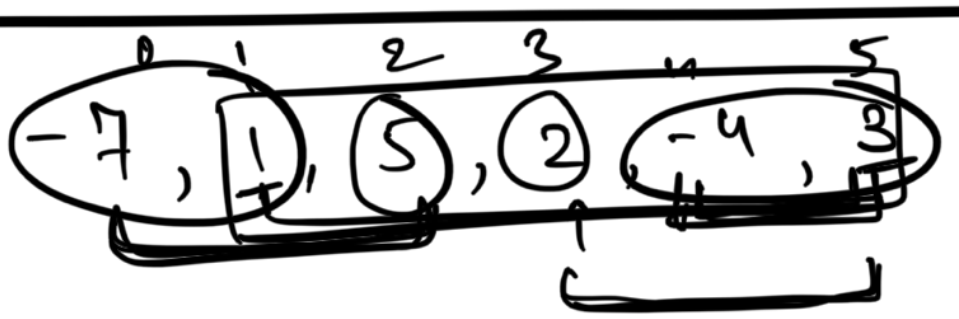
T.C ⇒ O(N)

S.C ⇒ O(N)

#3 Can we do it in O(1) S.C



$$\left.\begin{aligned} [sum'_L &= sum_L - a[i-1] \\ [sum'_R &= sum_R + a[i] \end{aligned}\right] \rightarrow O(1)$$



$i = 0$ ✗

$sum_L = 0$ ←

$sum_R = 1 + 5 + 2 + -4 + 3 = 7$

$i = 1$ ✗

$sum'_L = sum_L + a[0] = 0 + -7 = -7$

$\boxed{sum'_R} = 7 - 1 = 6.$ ✗

$i = 2$ ✗

$sum''_L = sum'_L + a[1] = -7 + 1 = -6$

$sum''_R = sum'_R - a[2] = 6 - 5 = 1$

$\boxed{i = 3}$

$sum_L = -6 + 5 = -1.$

$sum_R = 1 - 2 = -1$

Code:

```
# sum_L = 0 ✓
# sum_R = 0 ✓
for (i = 1 to n-1) sum_R += a[i] ✓
```

for ([redacted]

if(sum_L == sum_R) return '0';

for (i : 1 to n) {

sum_L += a[i-1]

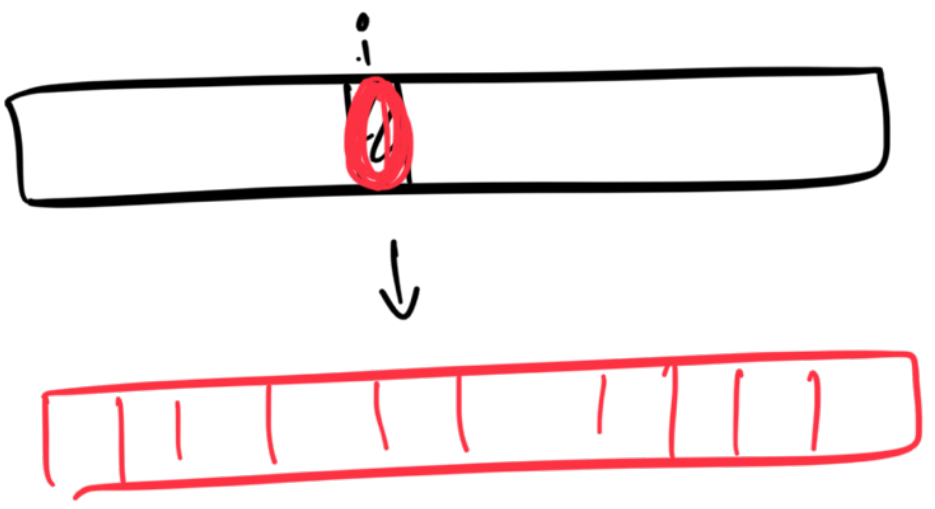
sum_R -= a[i]

if(sum_L == sum_R) return i;

}

T.C = O(N) S.C : O(1)

Q.2 : Count Special indexes :-



Sum of odd indices == Sum of even indices.

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

removing index '2'

	0	1	2	3
	2 ,	4 ,	3 ,	3

sum_{odd} = a[1] + a[3] = 4 + 3 = 7

sum_{even} = a[0] + a[2] = 2 + 3 = 5

	0	1	2	3	3
A :	2	4	2	3	3

A : 2, 4, 2, 3, 3

↓ remove '1' index

0	1	2	3
2,	2,	3,	3

A' :

$S_{odd} = 2 + 3 = 5$

$S_{even} = 2 + 3 = 5$

'1' is a special index

#1 Brute force :-

A :

2	4	2	3	3

$n-1$

0	1	2	3
2	4	3	3

A' :

for every (i)

O(N) { Iterate over A' array
find S_{odd}
S_{even}
Check if (S_{odd} == S_{even})
count++

T.C ⇒ $O(N^2)$ $N \to 10^5$

S.C ⇒ $O(N)$

#2 :- Optimized Approach

Observation

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

$S_o = 6+8$

$S_E = 5+7+9$

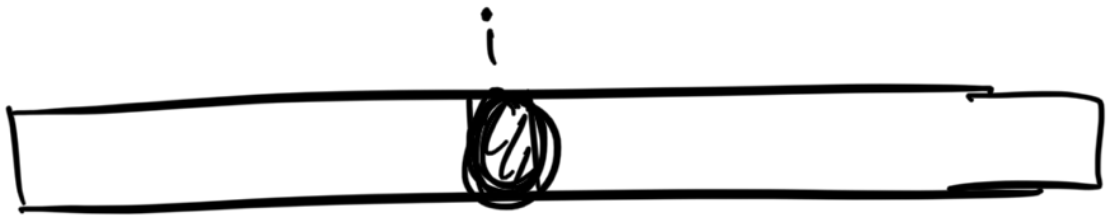
0	1	2	3	4	5	6	7
1	2	3	5	6	7	8	9

$S_o = 5+7+9$

$S_o = 6+8$

Right indexes are shifting from ODD to EVEN & EVEN to ODD.

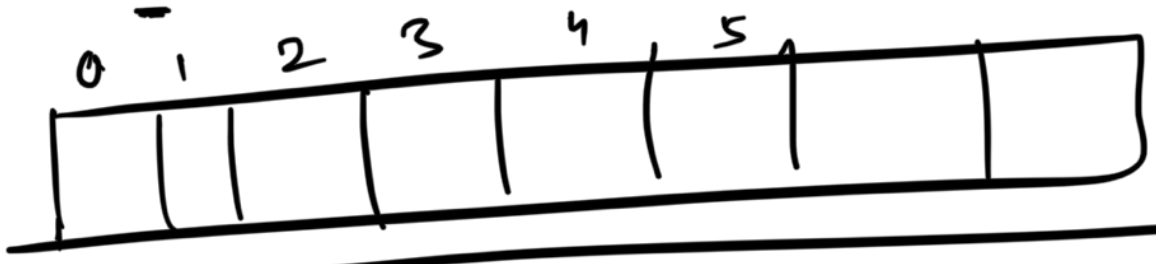
A :



$Sum_{L\,Even}$ $\quad$ $Sum_{R\,Even}$

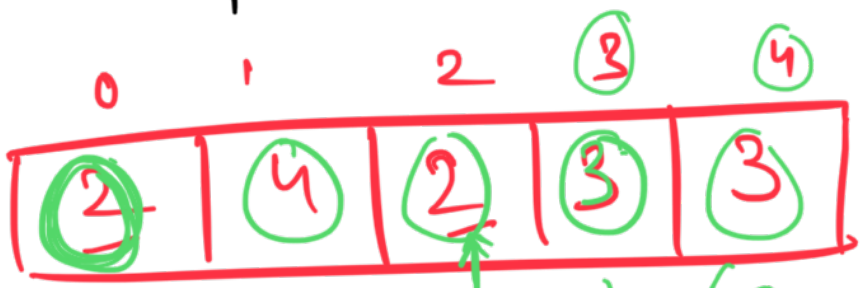
$Sum_{L\,odd}$ $\quad$ $Sum_{R\,odd}$

A'



$$Sum_{Even} = Sum_{L\,Even} + S_{R\,odd}$$
$$Sum_{odd} = Sum_{L\,odd} + S_{R\,Even}$$

A :

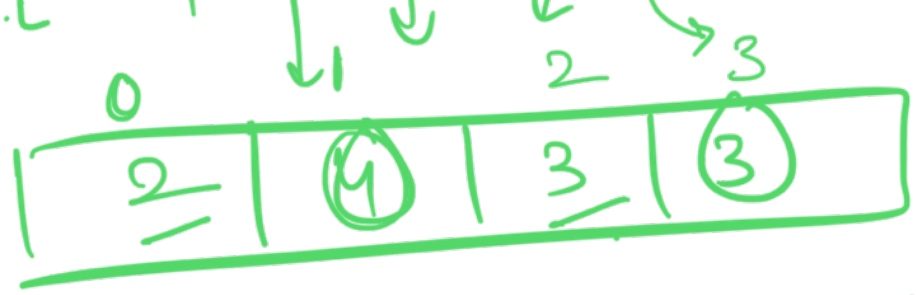


$S_{E.L} = 2$

$S_{O.L} = 4$

$S_{E.R} = 3$

$S_{O.R} = 3$

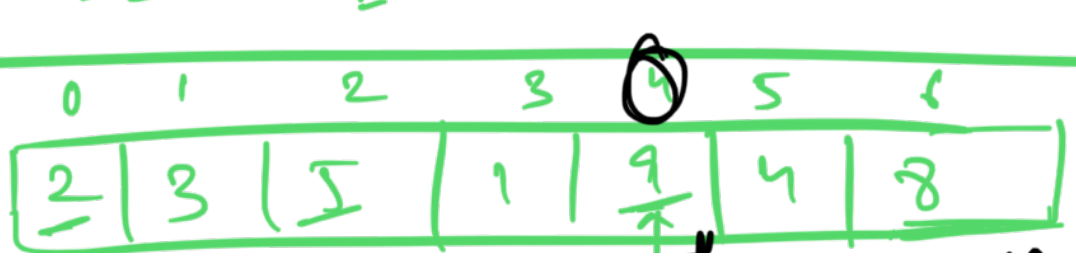


$S_E = 2+3$

$S_O = 4+3$

$$S_O = S_{O.L} + S_{E.R} = 4+3 = 7$$
$$S_E = S_{E.L} + S_{O.R} = 2+3 = 5$$

A :



$i = 4$

A' :

2	0	5	0	9	0	8

$P.S_E$:

0	1	2	3	4	5	6
2	2	7	7	16	16	24

A'' :

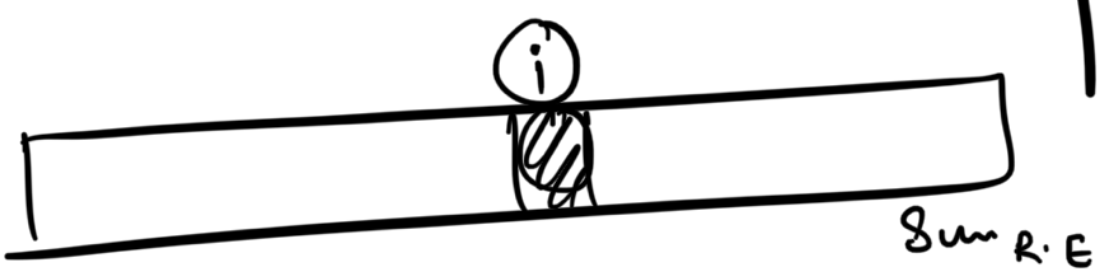
0	3	0	1	0	4	0

$P.S_O$

0	3	3	4	4	8	8

$S_E =$

$PS_E[i-1] + PS_O[n-1] - PS_O[4]$



$Sum_{L.E}$

$Sum_{L.O}$

$Sum_{R.O}$

* $Sum_{Even} = Sum_{L.E} + Sum_{R.O}$
* $Sum_{odd} = Sum_{L.O} + Sum_{R.E}$

for any index 'i' :

$$Sum_E = P.S_{Even}[i-1] + \left(PS_{odd}[n-1] - P.S_{odd}[i]\right)$$

$$Sum_O = P.S_{odd}[i-1] + \left(PS_{Even}[n-1] - PS_{Even}[i]\right)$$

T.C ⇒ $O(N)$

S.C ⇒ $O(N)$

Q.3 Consecutive Number Sum :-

Given an integer 'N', return number of ways it can be written as sum of distinct consecutive positive integers.

$N \le 10^{12}$

N = 5 :- 2 + 3 ⇒ 1

N = 9 : 4 + 5 , 2 + 3 + 4 ⇒ 2

N = 15 : 4 + 5 + 6 , 7 + 8 , 1 + 2 + 3 + 4 + 5 → 3

Brute force :-

Count = 0

1 + 2 + 3 + 4 + 5 ⇒ Count++ → 1

2 + 3 + 4 + 5 + 6 > 15 ✗

3 + 4 + 5 + 6 > 15 ✗

4 + 5 + 6 = 15 Count++ → 2

5 + 6 + 7 > 15

6 + 7 + 8 ✗

7 + 8 = 15 Count++ → 3

15 + 15 ✗

15 → Count++ 4

T.C ⇒ $O(N^2)$, → 10^8

for (i : 1 to N) →

Creating a series ←

#2 :- 2 pointer technique / Sliding window.

N = 15

1, 2, 3, 4, 5, 6, 7, 8 | 9, 10, 11, 12, 13, 14, 15

$\frac{N}{2}$, $\frac{N}{2}+1$

7.5

l = 1 , r = 1

sum = 1 < 15 , r++

sum = 1 + a[r] = 1 + 2 = 3 , r = 2 < 15

7+8

8+7

sum = 6 < 15,

sum = 10 < 15 , r++

[sum = 15 = 15, $\boxed{r = 5 \ \& \ l = 1}$ count++

↓ l++

sum = 15 − 1 = 14 < 15 , r++

sum = 20 , l = 2, r = 6

sum = 18, l = 3, r = 6

sum = 15, l = 4, 6 → count++

N → 10^{12}

T.C ⇒ O(N) → 10^{12}

S.C ⇒ O(1)

T.L.E

$$N = \underbrace{x + (x+1) + (x+2) + \cdots (x+K-1)}_{\text{'K' terms}}$$

$$N = K \cdot x + (1 + 2 + \cdots K-1)$$

$$= K \cdot x + \frac{K(K-1)}{2}$$

$$K \cdot x = N - \left(\frac{K \cdot (K-1)}{2}\right)$$

$$x = \frac{N - \frac{K(K-1)}{2}}{K}$$

$N = 15$ K ? : Number of terms in the series

$K = 1$ $\quad x = \frac{15}{1} = 15$ $\quad$ count ++

$K = 2$ $\quad x = \frac{15 - \frac{2 \cdot 1}{2}}{2} = \frac{15-1}{2} = 7$

$7 + 8 = 15$ $\quad$ count ++.

$K = 3$ $\quad x = \frac{15 - \frac{3 \cdot 2}{2}}{3} = 4$

$4 + 5 + 6$ $\quad$ count ++

$K = 4$ $\quad x = \frac{15 - \frac{4 \cdot 3}{2}}{4} = \frac{9}{4}$

$K = 5$ $\qquad x = \dfrac{15 - \frac{4 \cdot 3}{2}}{4} = 1$

$1 + 2 + 3 + 4 + 5 = 15$

Count++ $\quad 2$

$$\text{Sum} = x + (x+1) + (x+3) + \cdots (x + (K-1))$$

$$= K \cdot x + (\underbrace{1 + 2 + 3 + \cdots (K-1)})$$

$$\text{Sum} = K \cdot x + \frac{K \cdot (K-1)}{2}$$

Pseudo Code :

```
K = 1
while ( N - K·(K-1)/2 > 0) {
    if ( N - (K·(K-1))/2 % K == 0)
        Count++;

    K++;
}
return Count;
```

T.C $\left[N - \dfrac{K \cdot (K-1)}{2} > 0 \right.$

$N > K \cdot (K-1)$

$$\frac{\quad}{2}$$

$$K \cdot (K-1) < 2N$$

$$(K-1)(K-1) < 2N$$

$$(K-1)^2 < 2N$$

$$K-1 < \sqrt{2N}$$

$$\boxed{K < \sqrt{2N} + 1}$$

$$T.C \Rightarrow O(\sqrt{2N} + 1)$$

$$\Rightarrow O(\sqrt{N})$$

Q.4 Given an array 'A' of size N, find a subarray of size (K) with least average.

A : [10, 20, 30, 5, 3, 2, 1]

$N = 7$
$K = 3$

$$\text{avg} = \frac{\text{sum of K elements}}{K}$$

10, 20, 30 → 60
20, 30, 5 → 55
30, 5, 3 → 38
5, 3, 2 → 10
3, 2, 1 → 6

$\boxed{\text{avg} \propto \text{sum}}$

Subarray of size (K) with minimum sum.

Brute force :-

0	1	2	3	4	5	6
10	20	30	5	8	2	1

T.C :- $O(n \cdot K)$

2

2, 6, 8, 3, 1, 4

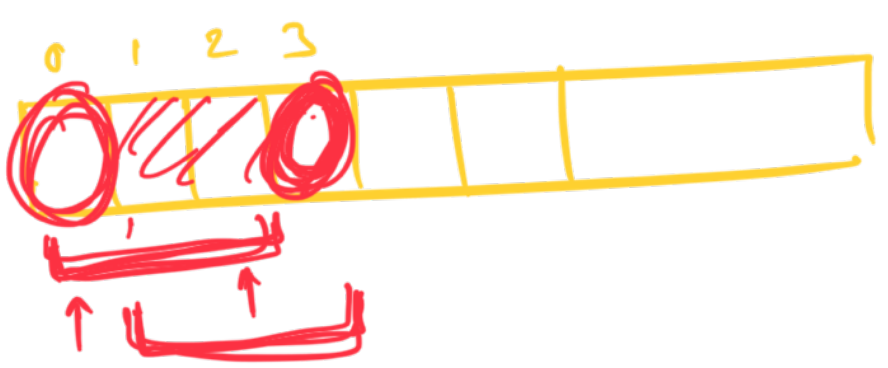
$sum_1 = 2 + 6 + 8 = 16.$

$sum_2 = sum_1 - a[0] + a[3]$ #

$= 16 - 2 + 3 = 17$

$sum_3 = sum_2 - 6 + 1 = 12$

$sum_4 = sum_3 - 8 + 4 = 8$



$\rightarrow \{0, 1, 2\}$

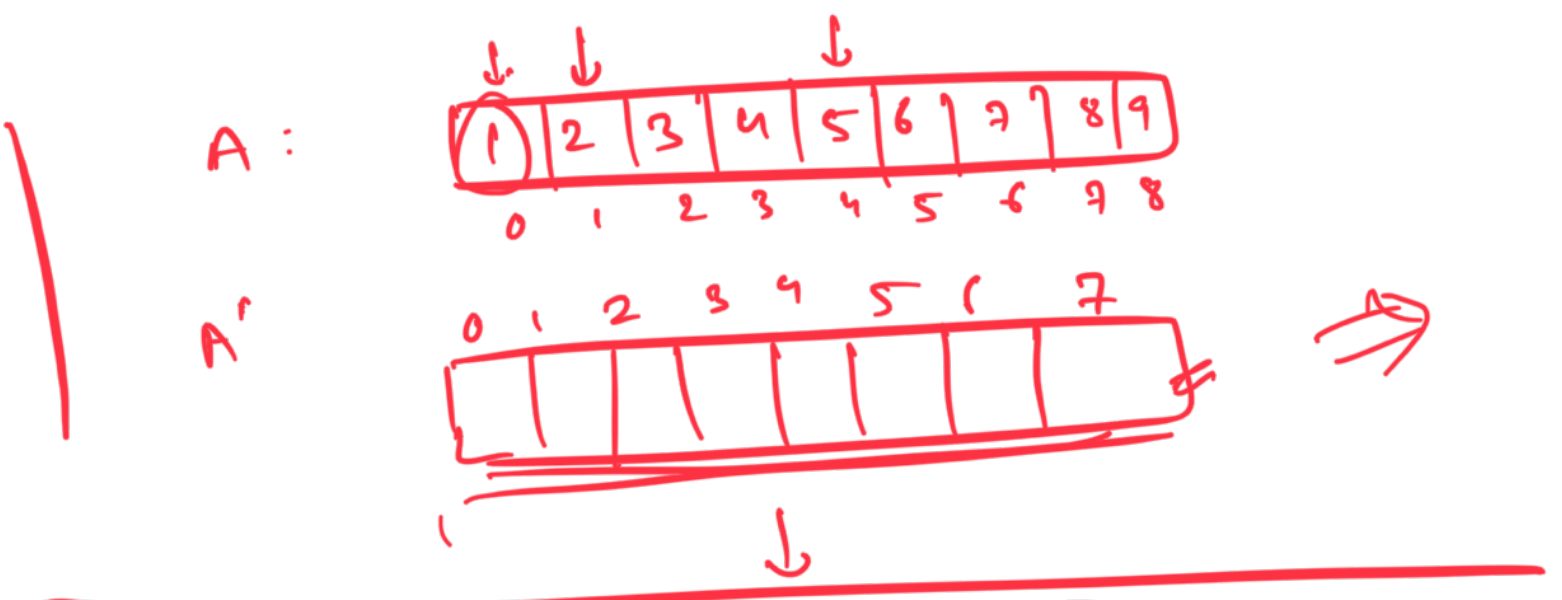
$\Rightarrow$

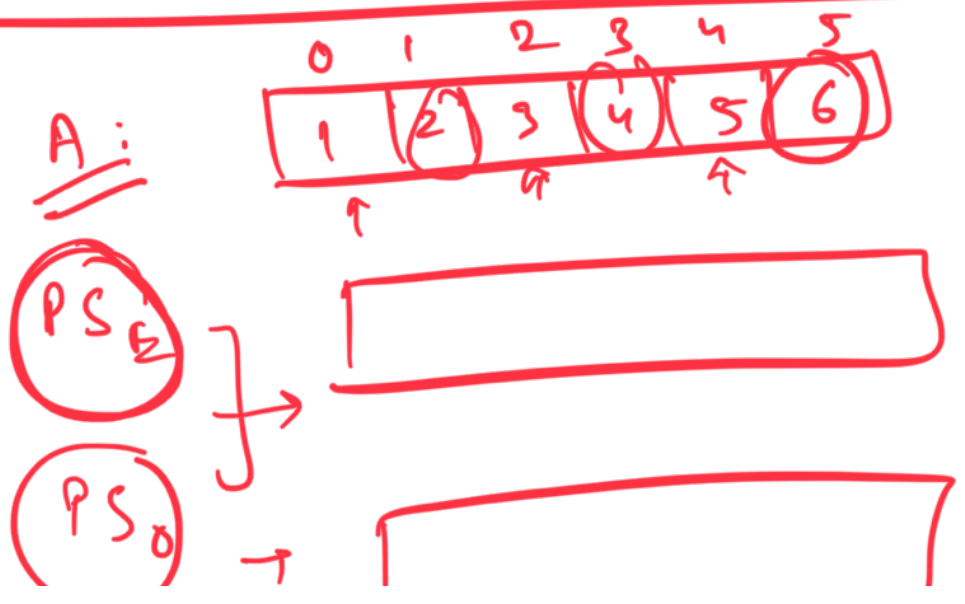
```
sum = 0
for (i = 0 to k-1) {
```

```
        sum += a[i]
    }
    ans = sum;
    for (i = 1; i < n-k; i++) {
        sum -= a[i-1]
        sum += a[i+k-1]
        ans = min(sum, ans)
    }
    return ans/k;
```

T.C ⇒ O(N)

S.C ⇒ O(1)

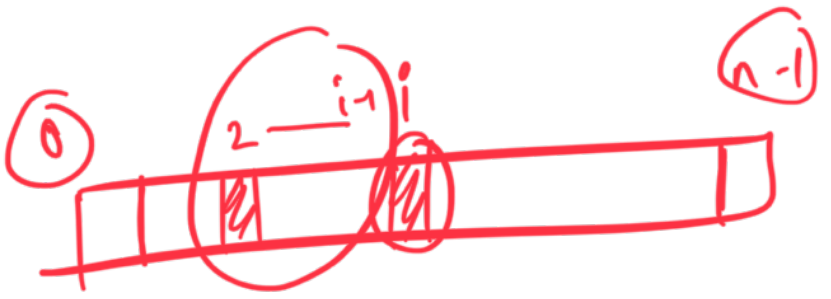




2*N

4*N

⇒ O(N)



$Sum_L = Sum_R$

$\downarrow$ $sum(0 \ldots i-1)$

$\rightarrow$ $sum(i+1 \ldots n-1)$