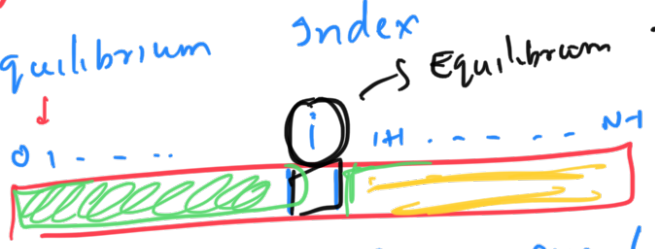
Problem - Solving -2

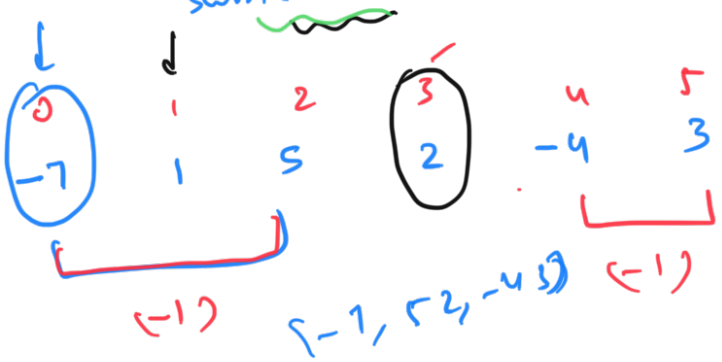
Question:

Array of size N.

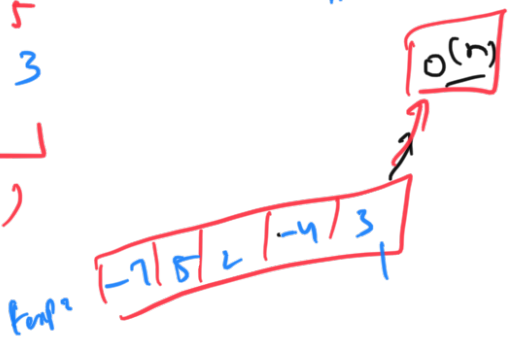
Equilibrium Index



sum(0 ... i-1) = sum(i+1 .... n-1)



Return any index



Brute Force : n

For every 'i'

-> Iterate to Left O(n)
-> Iterate to Right O(n)

T.C: $O(n*n) = O(n^2)$

S.C: O(1)

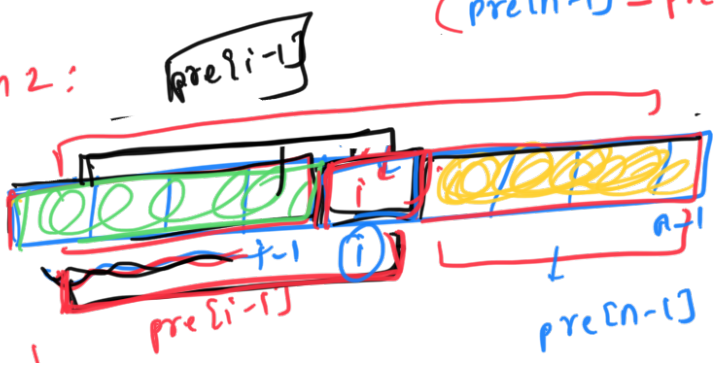
leftSum = pre[i-1]

(pre[n-1] - pre[i])

prefix

sum(0 ... i-1)

Approach 2:



Arr

pre

For any 'i',

if (pre[i-1] == pre[n-1] - pre[i])

Eq Index!

pre[i] = sum[0...i]

A = -7 1 5 2 -4 3

Pre = -7 -6 -1 1 -3 0 ⇒ T.C: O(n)

0 1 2 3 4 5

pre[2] = sum(0...2)

pre[i] = sum(0...i)

if[pre[n-1] - pre[0] == 0) return 0;

for (i=1; i<n; i++) {

if (pre[i-1] == pre[n-1] - pre[i])

return i;

}

return -1;

-7 1 5 2 -4 3

0 1 2 3 4 5

5 1 2 3 -6

0 1 2 3 4

Special case

Left = 0

right = 0

T.C: O(n)

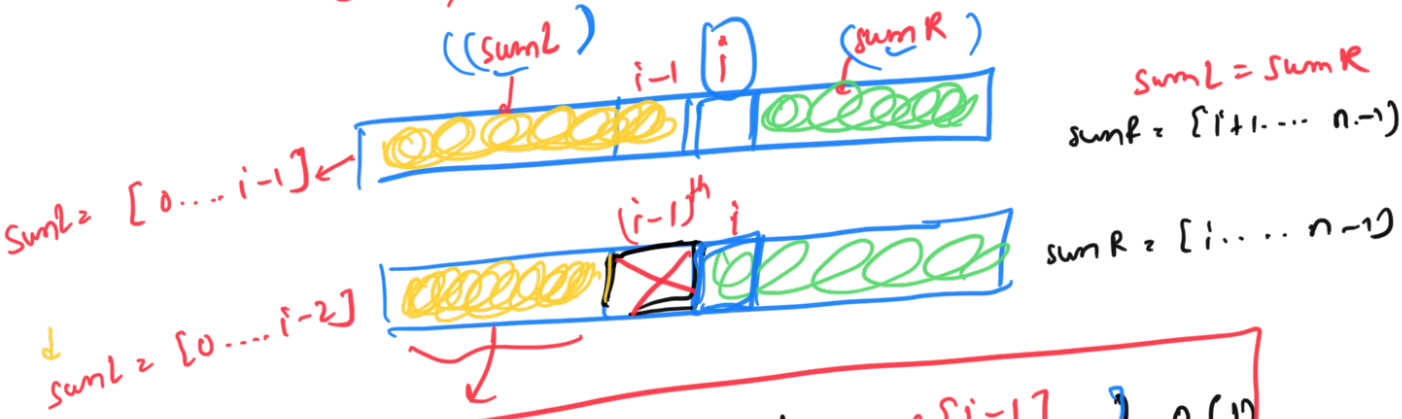
S.C: O(n)

& Prefix Sum Array

Approach 3:

LeftSum, rightSum

L / R




```
{ sumL = sumL - a[i-1]
  sumR = sumR + a[i]   }  O(1)
```

A = [-1, 1, 5, 3, 2, 4, -4, 5, 3] (indices 0 1 2 3 4)

Ans = 3

sumL = sum(0 .... n-2) = -3

sumR = 0

sumL = -3 - (-4) = 1

sumR = 3

(Lecture Notes)

T.C : O(n)

S.C : O(1)

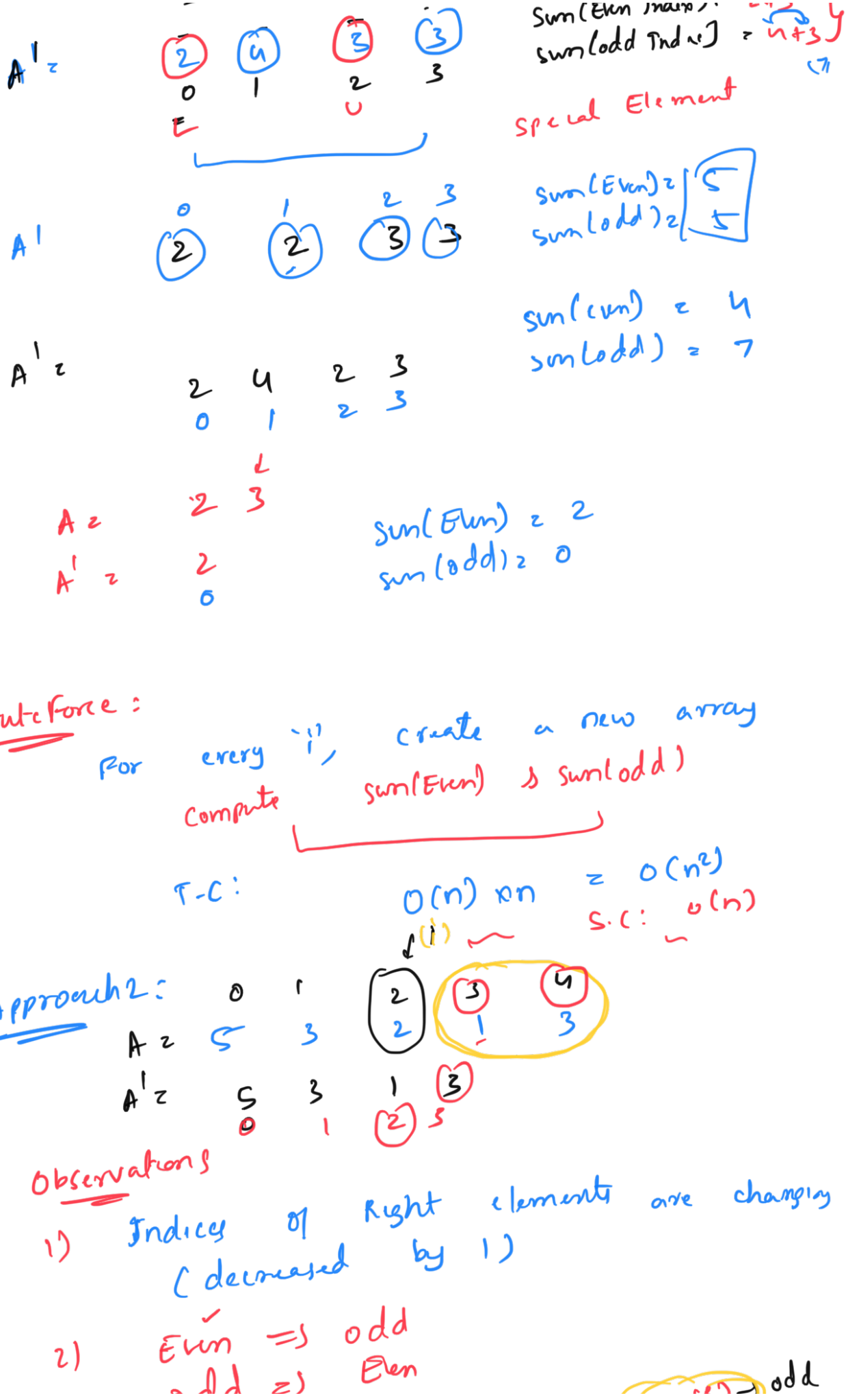
Question: Count special Elements

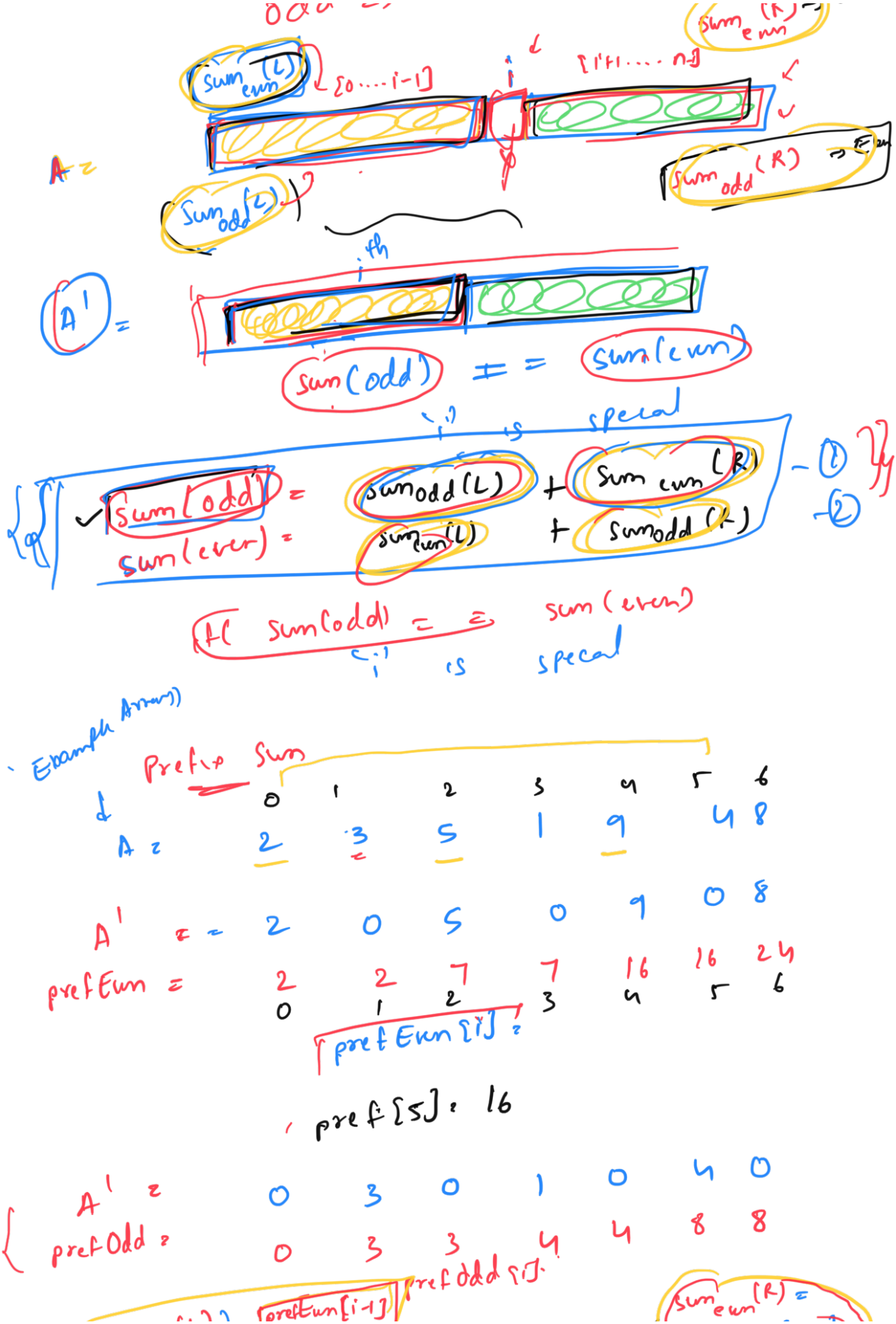
Array of size 'n', Count no of special elements

Ans = 1

special ele.

A = [1, 1, 4, 2, 2, 3, 3, 4] (indices 0 2)


A' =
2 4 3 3
0 1 2 3
E O
special Element
sum(Even) = 5
sum(odd) = 5
sum(even) = 4
sum(odd) = 7
A =
A' =
sum(Even) = 2
sum(odd) = 0
Brute Force :
For every 'i', create a new array
Compute sum(Even) & sum(odd)
T.C : O(n) * n = O(n²)
S.C : O(n)
Approach 2 :
Observations
1) Indices of Right elements are changing (decreased by 1)
2) Even => odd
odd => Even


A =
$sum_{even}(L)$
[0 .... i-1]
i
[i+1 .... n-1]
$sum_{even}(R)$
$sum_{odd}(L)$
$sum_{odd}(R)$
A' =
ith
sum(odd) == sum(even)
i is special
sum(odd) = $sum_{odd}(L)$ + $sum_{even}(R)$ - (1)
sum(even) = $sum_{even}(L)$ + $sum_{odd}(R)$ - (2)
if sum(odd) == sum(even)
i is special
Example Array)
Prefix Sum
0 1 2 3 4 5 6
A = 2 3 5 1 9 4 8
A' = 2 0 5 0 9 0 8
prefEven = 2 2 7 7 16 16 24
0 1 2 3 4 5 6
prefEven[i]
pref[5] = 16
A' = 0 3 0 1 0 4 0
prefOdd = 0 3 3 4 4 8 8
prefEven[i-1]
prefOdd[i]
$sum_{even}(R)$ =

$sum_{even}^{(L)}$ + prefEven[n-1] + prefEven[i]

$sum_{odd}^{(R)}$

$(sum_{odd}^{(L)})$ = prefOdd[i-1]

prefOdd[n-1] − prefOdd[i]

For any 'i'

if(prefEven[i-1] + prefOdd[n-1] − prefOdd[i]

== prefOdd[i-1] + prefEven[n-1] − prefEven[i])

{

'i' is special (i = 0)

(i=0)

for(i=1; i<n; i++) {

if(prefEven[i-1] ...)

} // check for 0th Element

Corner Case // i = 0

T.C : O(n)

S.C : O(n)

Question : Consecutive Numbers Sum

Given Integer N (10^{12})

ways it can written as

Return no. of consecutive positive ... of a sum

N = 5 (2+3), (5) (Ans = 2)

N = 9 (4+5), (2+3+4), 9 } Ans = 3

N = 15 7+8, 1+2+3+4+5, 4+5+6, 15 } Ans = 4

Brute force: N = 15

1 + 2 + 3 + 4 + 5

2 + 3 + 4 + 5 + 6 > 15

3 + 4 + 5 + 6 > 15

4 + 5 + 6 = 15

count++;

count++;

T.C: (N x N)

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

S.C: O(1)

[1 ... N] as starting point

N = 10^{12}

$O(N^2)$ 10^{24}

Approach 2:

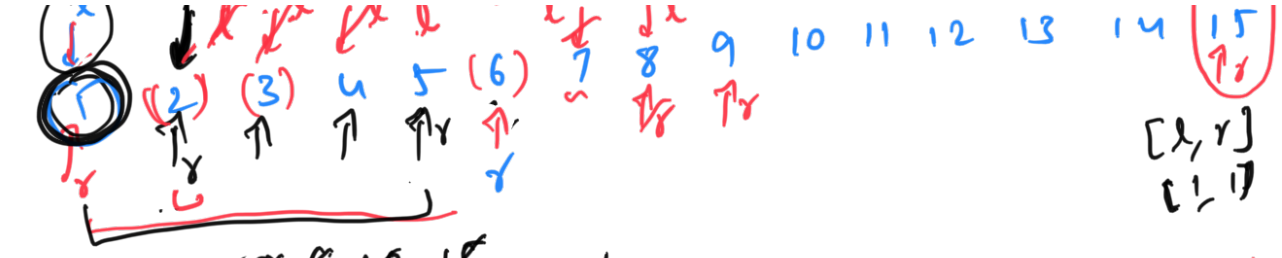
(2 Pointer / Sliding Window Technique)

N = 15

[1+2+3+4+5, 4+5+6, 7+8, 15]

Ans 2 4

[4+5+6] = 15



curr_sum = 15 20 18
15 15 15 8 17

15 - a[i]
15 - 10

(1+2+3+4+5)

L = 4
R = 6

T.C :

L : 1 → N
R : 1 → N

L = 1 , L = N
R = 1 R = N

(2N) operations
= T.C : O(N)

N = 10^{12}

while (l ≤ R)

N = 15

(15)

$(\frac{N}{2})$

$O(\frac{N}{2}) = O(N)$

$(\frac{N}{2})$

(3)

(5+6)

length 2

$(\frac{N}{2})$

$O_2(n)$

A.P where d = 1

a+(n-1)d

$N = 10^{12}$

n+n+n. -- k times

Efficient Approach :

(K terms)

$N = \underbrace{x}_{1^{st}} + \underbrace{(x+1)}_{2^{nd}} + \underbrace{(x+2)}_{3^{rd}} + \underbrace{(x+3)}_{4^{th}} \ldots$

$[x + (k-1)]^{th}$

$(1+2+3+\ldots+k-1)$

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

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

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

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

x = First terms of Series

K = No. of terms

$N = 15$

$\{15\} = K = 1$

$N = 15$

K is atleast 1

$K = 1$ $\quad x = \frac{\left(15 - \frac{1(0)}{2}\right)}{1} = 15 \quad \{15\}$

$K = 2$ $\quad x = \frac{\left(15 - \frac{2(1)}{2}\right)}{2} = 7 \quad \{7, 8\}$

$K = 3$ $\quad x = \frac{\left(15 - \frac{3(2)}{2}\right)}{3} = 4 = \{4, 5, 6\}$

num/den = 0

$K = 4$ $\quad x = \frac{\left(15 - \frac{4 \times 3}{2}\right)}{4} = \left(\frac{9}{4}\right)$ ✗

$K = 5$ $\quad x = \frac{\left(15 - \frac{5 \times 4}{2}\right)}{5}, \frac{5}{5} = 1 = \{1, 2, 3, 4, 5\}$

(num/den = 0)

$K = 6$ $\quad x = \frac{\left(15 - \frac{6 \times 5}{2}\right)}{6} = \frac{0}{6} = 0$

$K = 7$ $\quad = x = \frac{\left(15 - \frac{7 \cdot 6}{2}\right)}{7} = \frac{15 - 21}{7} = \frac{-6}{21}$

```
k=1
while ( N - k(k-1)/2 > 0 ) {
    if ( [N - k(k-1)/2] % k == 0 ) {
        count++;
    }
    k++;
}
```

$k(k-1) < 2N$

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

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

$(k-1) < \sqrt{2N}$

$k < \sqrt{2N} + 1$

k : 1 → $(\sqrt{2N} + 1)$

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

C.P

A → N

Question: Find a subarray of size 'k' with the least average

k=3
N=7

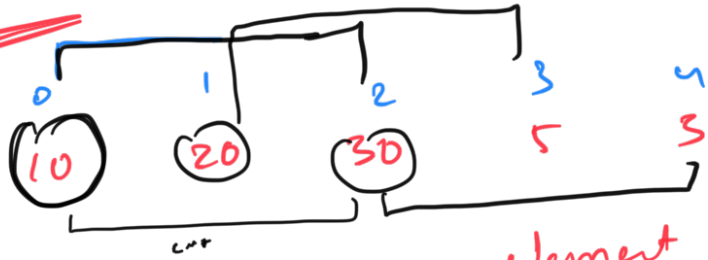
0	1	2	3	4	5	6
			5	3	2	1

A = 10 20 30

average = sum / (K) → constant

Q: Find subarray of size 'k' with least average ≈ sum

Brute Force



k = 3

ans = 60

→ Consider every element as start index.

T.C: O(n * K)

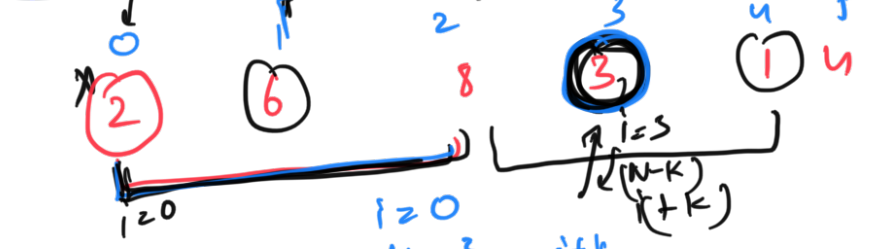
T.C: O(n.K)

Sliding Window / 2 Pointer

N = 6

k = 3

(3+3) = 6



A =

ans = INT_MAX

sum = 16 _

{i ... i+k-1} i+k

sum = 16 + 3 - 2 = 17

sum = 17 + 1 - 6 = 12

sum = 12 + 4 - 8 = 8

sum = 0;

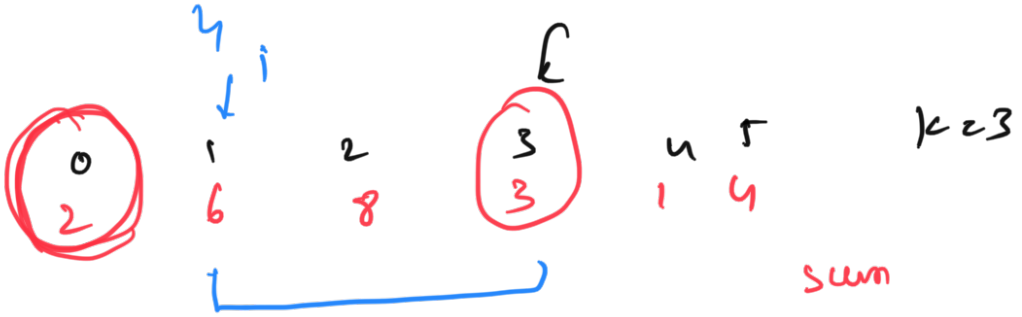
for(i = 1;

for(int i=0; i<k; i++)
    sum += a[i];

ans = sum;

for(i=0; i<n-k; i++) {
    sum += a[i+k];
    sum -= a[i];
    ans = min(ans, sum);
}

for(i=1; i<_____; i++) {
    sum -= a[i-1];
    sum += a[i+k-1];
}



sum = [0...k-1] [i...i+k-1]

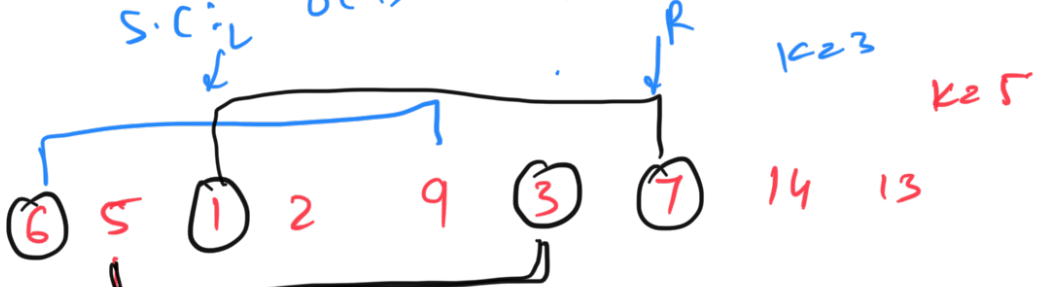
i=1, k=3

i+k = (4)-1

for(i=k; i<n...

T.C: O(n)

S.C: O(1) → O(n)



A=

Sum = 23

Sum = 23 + 3 − 6 = 20

sum = 20 + 7 − 1

$$n = \left\lfloor \frac{N - \frac{k(k-1)}{2}}{k} \right\rfloor$$

n, k have to be integers

N = 15

$$k = 4 \qquad \left\lfloor \frac{15 - \frac{4 \cdot 3}{2}}{4} \right\rfloor = \left\lfloor \frac{9}{4} \right\rfloor$$

num = 9
denomnt = 4

9 % 4

$\left(\frac{num}{dem}\right)$

(num % dem == 0)

num %

for (i = 1 ; i < n)

[1 ... n-1]

3 + 4 + 5 + 6 = 18

N = 15

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

[l, r]

curr_sum = 18 − 15 − 1 = 20 − 18 = 2

count = 1

L=2, R=6

Equilibrium Index : O(1) space

```
for(int i = 0; i < n-1; i++)
      sumL += a[i];
sumR = 0;
if(sumL == 0)
      last element is the equilibrium index

for(i = n-2; i>=0; i--){
      sumL -= a[i];
      sumR += a[i+1];
      if(sumL == sumR)
            return i;
}
```

Equilibrium Index : O(n) space

```
// Generate Prefix Sum
if(pref[n-1] - pref[0] == 0) return 0;  // i = 0
for(i = 1; i < n; i++)
      if(pref[i-1] == pref[n - 1] - pref[i])
            return i;
```

Subarray with least average

```
for(int i = 0; i < k; i++)
      sum += a[i];
min = sum;
for(int i = 0; i < n-k; i++){
      sum -= a[i];
      sum += a[i+k];
      min = min(sum, min);
}
return min/k
```

Consecutive Numbers Sum

```
k = 1;
while(A - k(k-1)/2 > 0){
      if((A - k(k-1)/2)%k == 0)
            count++;
      k++;
}
```