⊛ Prefix Sums

Given an array of N elements & Q queries!
For each query, calculate the sum of all elements in the index range $[L, R]$ $\quad [L, R]$

$0 <= L <= R < N$

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

L	R	Sum
4	8	9
6	9	3
0	4	14
7	7	-9

```
for (j=1; j<=Q; j++) {          // Q
    // [L, R]
    sum = 0;
    for (i=L; i<=R; i++) {      // O(N)
        sum += A[i];
    }
    // ans of this query -> sum
}
```

$1 Q \rightarrow O(N)$

$Q Q's \rightarrow O(Q \cdot N)$

$SC: O(1)$

$1 <= N <= 10^5$

$1 <= Q <= 10^5$

$10^5 \cdot 10^5 = 10^{10}$ ops

10^8 ops → 1 sec

10^{10} ops ↓ 100 sec!

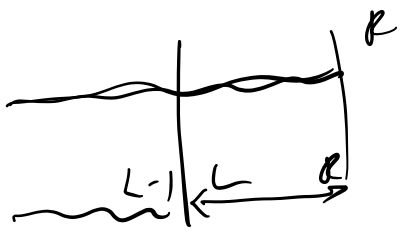
- Given the score of the first 10 overs of batting!
  After every over, current score → given

Overs:	1	2	3	4	5	6	7	8	9	10
Score:	2	8	14	29	31	49	65	79	88	97

- Total runs scored in the 10th over? [10, 10]
  $= S[10] - S[9] =$
  $97 - 88 = 9$ //

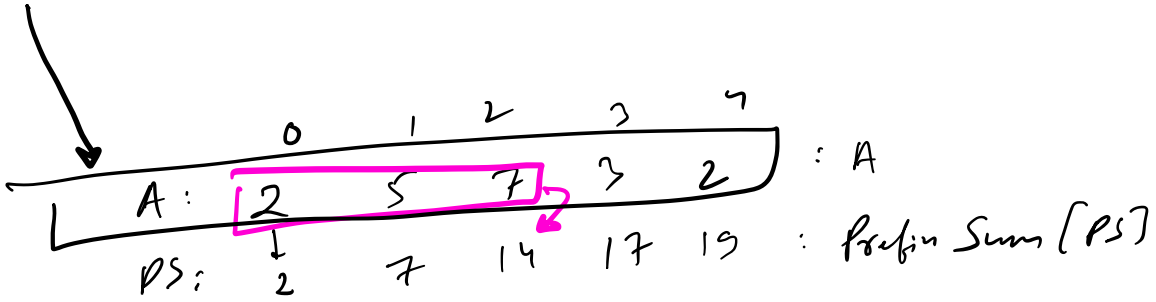
- _____ Last 5 overs? [6, 10]
  $= S[10] - S[5]$
  $= 97 - 31 = 66$ //

- _____ [3, 6] =
  $S[6] - S[2] = 49 - 8$
  $= 41$ //



Score (L, R): $S[R] - S[L-1]$

⊙ Cumulative Sum from start = Prefix Sums



$PS[4] \rightarrow sum[0, 4]$

$PS[i] \rightarrow sum[0, i]$

	0	1	2	3 (L-1)	4 (L)	5	6	7	8 (R)	9
A:	-3	6	2	4	5	2	8	-5	3	1
PS:	-3	3	5	9	14	16	24	19	18	19

[4 - 8] : $sum[0, 8] = sum[0, 3] + sum[4, 8]$

$PS[8] = PS[3] + sum[4, 8]$

$sum[4, 8] = PS[8] - PS[3]$

$sum[L, R] = PS[R] - PS[L-1]$

$[4, 8) = PS[8] - PS[3]$

$= 18 - 9 = 9$

L == 0

$Sum[0, 3] = PS[3] - PS[-1]$

$= PS[3]$

$Sum[0, i] = PS[i]$

Given the arr A. Build the PS Array!

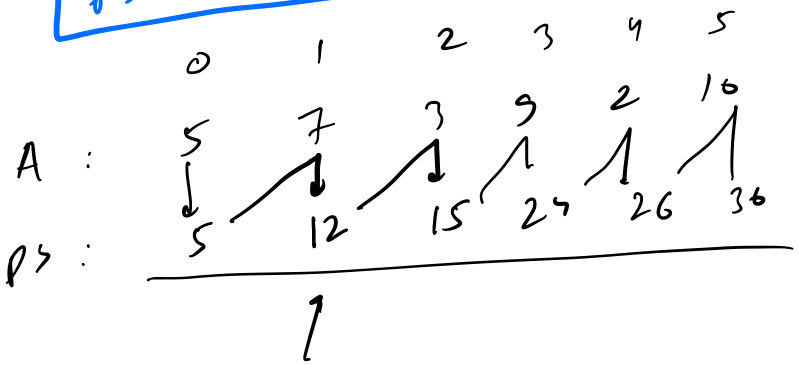
$PS[0] = A[0]$

$PS[1] = A[0] + A[1] : PS[0] + A[1]$

$PS[2] = A[0] + A[1] + A[2] = PS[1] + A[2]$

$PS[3] = A[0] + A[1] + A[2] + A[3] = PS[2] + A[3]$

$PS[i] = PS[i-1] + A[i]$



```
// A[N];
int ps[N];
ps[0] = A[0];
for(i=1; i<N; i++) {
    ps[i] = ps[i-1] + A[i];
}

for(i=1; i<=q; i++) {
    // [L, R]
    if(L==0)
        sum = ps[R]
    else
        sum = ps[R] - ps[L-1];
}
```

O(N) + O(q)

TC : O(N+q)

SC : O(N) — O(1)

A : 1 3 5 9

We may use the same array!

WAY 1 (BF)

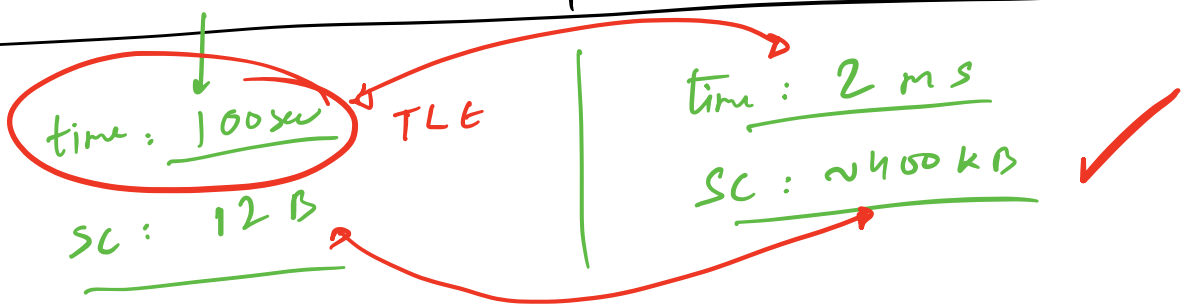
TC : O(q.N)

SC : O(1)

WAY 2 (PS)

TC : O(q+N)

SC : O(N)



Q Equilibrium Index (EI)

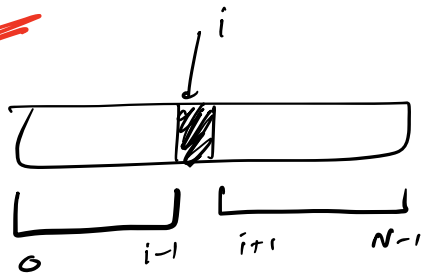
Given an array of N elements.
Count the no. of Equilibrium Index (EI).

EI → if sum of all elements before it
= sum of all elements after it!

```
cnt = 0;
for (i = 0; i < N; i++)
    SL = 0;
    for (j = 0; j <= i-1; j++) {
        SL += A[j];
    }
    SR = 0
    for (j = i+1; j <= N-1; j++) {
        SR += A[j];
    }
    if ( SL == SR) {
        cnt++;
    }
}
```

N

i



i

N-i-1

TC: O(N²)

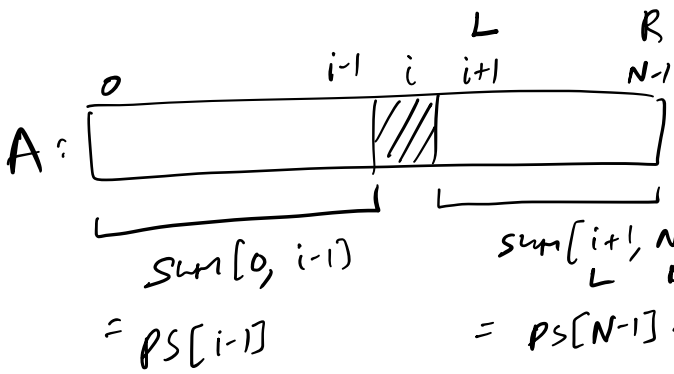
SC: O(1)

PS[R] - PS[L-1]

PS[N-1] - PS[i+1-1]

PS[N-1] - PS[i]

II)



$PS_R - PS_{L-1}$

Algo

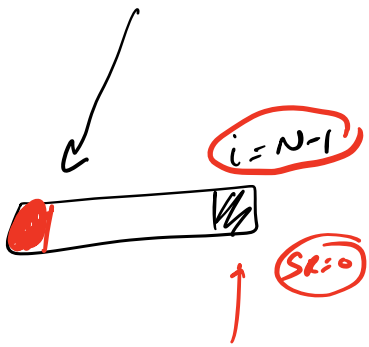
1) Build PS Array! → O(N)

2)

```
cnt = 0;
for (i = 0; i < N; i++) {
    if (i == 0) {
        SL = 0
    }
    else {
        SL = PS[i-1]
    }

    SR = PS[N-1] - PS[i];
    if (SL == SR) {
        cnt++;
    }
}
```

→ O(N)

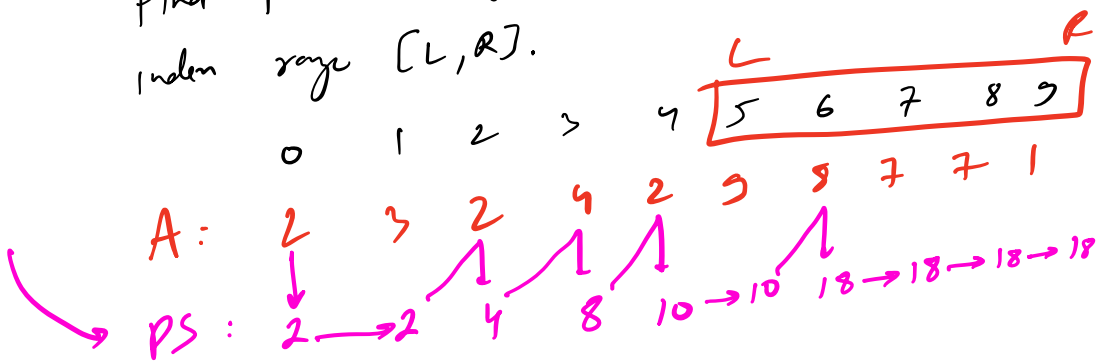


TC = O(N) + O(N)

TC = O(N)

SC = O(N) → O(1)

Q Given an array of size N.
Given Q queries → [L, R]
Find the sum of all EVEN elements in the index range [L, R].



L	R	Sum
0	5	10
5	9	0
5	9	8

$$PS[i] = PS[i-1] + \begin{cases} \text{if } (A[i]\%2==0) \rightarrow A[i] \\ \text{else} \rightarrow 0 \end{cases}$$

idea: Consider odd no's as 0.

```
PS[N];
PS[0] = (A[0]%2==0 ? A[0] : 0);

for(i=1; i<N; i++){
    PS[i] = PS[i-1] + (A[i]%2==0 ? A[i] : 0);
}
```

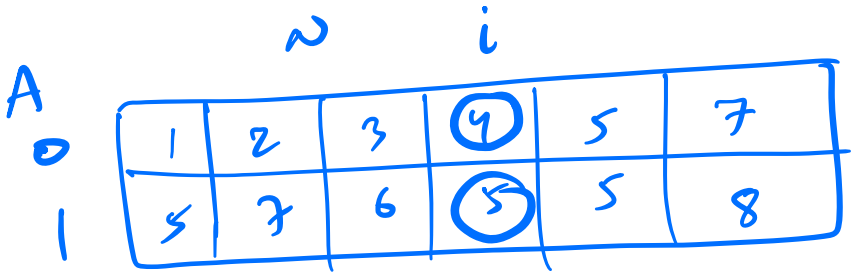
O(N)

```
for(i = 1; i <= q; i++) {
    // [L, R]
    if(L == 0)
        sum = ps[R];
    else
        sum = ps[R] - ps[L-1];
}
```

→ O(q)

TC: O(N+q)

SC: O(N)



(2 × N)

A[0][i] → L
A[1][i] → R