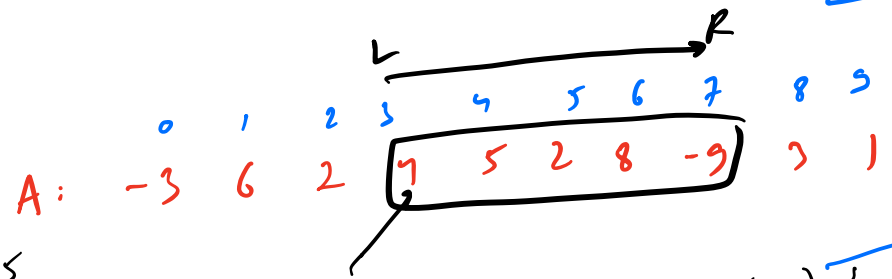
⊙ Prefix Sums

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

$0 <= L <= R < N$



Q = 5

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

```
for(i=1; i<=Q; i++) {
    // [L, R]
    sum = 0;
    for(j=L; j<=R; j++) {
        sum += A[j];
    }
    // ans of the query → sum.
}
```

Q → $O(N)$

1 Qu → $O(N)$

Q Qu → $O(Q \cdot N)$

TC = $O(QN)$

SC = $O(1)$

$1 <= N <= 10^6$

$1 <= Q <= 10^5$

ops → $10^5 \times 10^6 = 10^{11}$

10^8 ops → 1 sec

10^{11} ops → 10^3 sec

TLE!

⊙ 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	23	31	43	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]

Run:	3	5	2	0	5	3
Score:	3	8	10	10	15	18

$S[6] - S[2]$

$49 - 8$

$= 41$ //

runs [L, R] = S[R] − S[L − 1]

⊙ Cumulative sum for start → Prefix Sum

	0	1	2	3	4
A:	2	5	7	3	2
PS:	2	7	14	17	19

PS[3] → sum [0, 3]

PS[i] → sum [0, i]

	0	1	2	3	4	5	6	7	8	9
A:	-3	6	2	4	5	2	8	-9	3	1
PS:	-3	3	5	9	14	16	24	15	18	19

Sum [4, 8]: sum[0, 8] = sum [0, 3] + sum [4, 8]

sum [4, 8] = sum [0, 8] − sum [0, 3]

Sum [4, 8] = PS[8] − PS[3]

Sum [L, R] = PS[R] − PS[L−1]

Eg: sum [4, 8] = PS[8] − PS[3]

18 − 9 = 9 //

L == 0

sum[0, 3] = PS[3] − PS[-1]

PS[-1] → 0

sum(0, i) → PS[i]

Q Given an Array A. Build the PS array!

PS[0] = A[0]

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

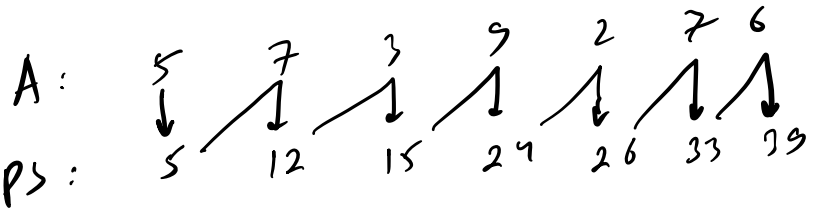
PS[2] = A[0] + A[1] + A[2]

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

PS[0] → A[0]

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

i > 0

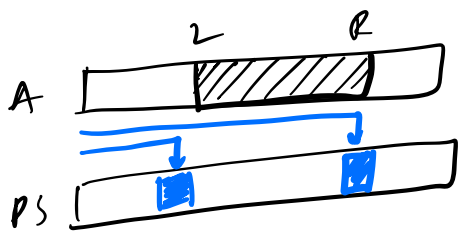


1. Build the PS[]
2. Ans queries using PS[].

```
// A[], N, Q, ...

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];

    // sum → Ans of the ith query!
}
```

N × O(1) : O(N)

Q × O(1) : O(Q)



TC : O(N+Q)

SC : O(N)

PS[]

A: 2 5 10 12 13

2 3 5 2 1

O(1)

① Immutable A[] : SC O(N)

② Restore it back! ✓ SC : O(1)

WAY 1

TC: O(NQ)

#ops → $10^6 \times 10^5$

10^{11}

$\sim 10^3$ sec

$1 <= N <= 10^6$

$1 <= Q <= 10^5$

WAY 2

TC: O(N+Q)

$10^6 + 10^5$

$\sim 10^6$ ops

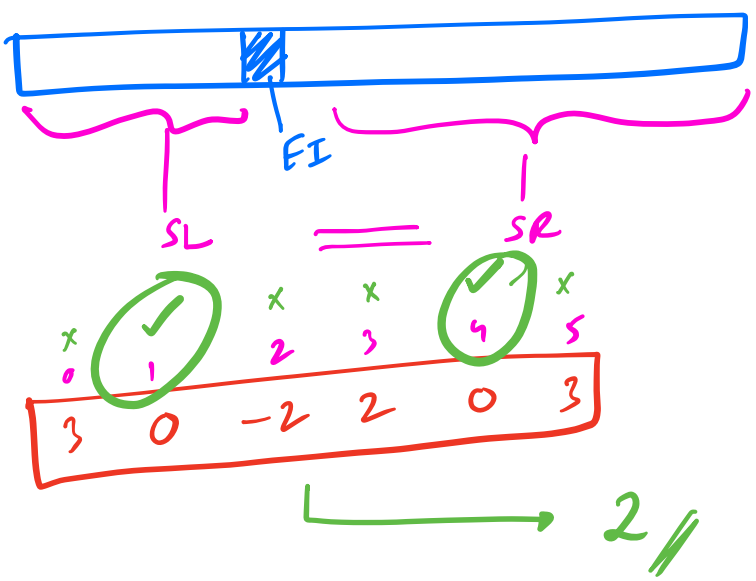
$\sim 10^{-2}$ sec

Q Equilibrium Index → [EI]

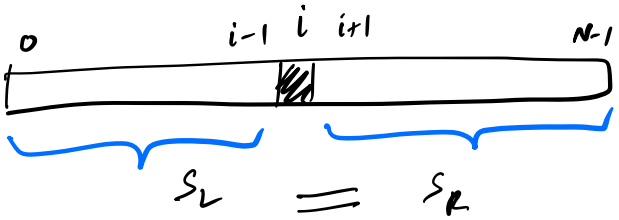
Given an array of N elements.

Count the no. of EI.

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



I) BF



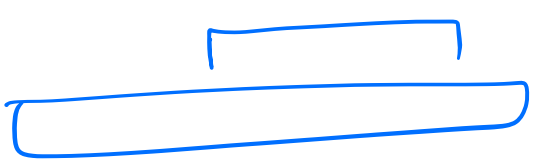

```
cnt = 0;
for(i=0; i<N; i++){
    SL = 0;
    for(j=0; j<i; j++){
        SL += A[j];
    }
    SR = 0
    for(j=i+1; j<N; j++){
        SR += A[j]
    if(SL == SR){
        cnt++;
    }
}
retcnt;
```

N

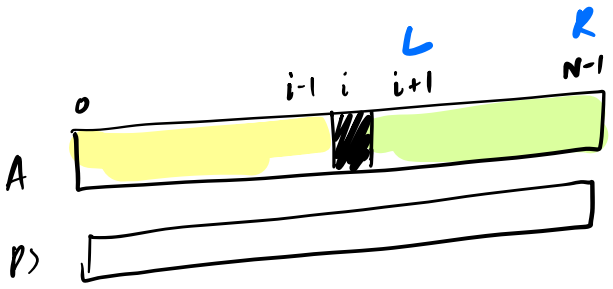
O(N)

TC = $O(N^2)$

SC = $O(1)$



II)



S_L = sum [0, i-1] → PS[i-1]

S_R = sum [i+1, N-1] → PS[N-1] - PS[i]

BUILD the PS[] → O(N)

```
cnt = 0;
for (i = 0; i < N; i++) {
    S_L = (i == 0 ? 0 : PS[i-1]);
    S_R = PS[N-1] - PS[i]
    if (S_L == S_R) {
        cnt++;
    }
}
ret cnt;
```

N x O(1) = O(N)

O(1)

2N

TC = O(N)

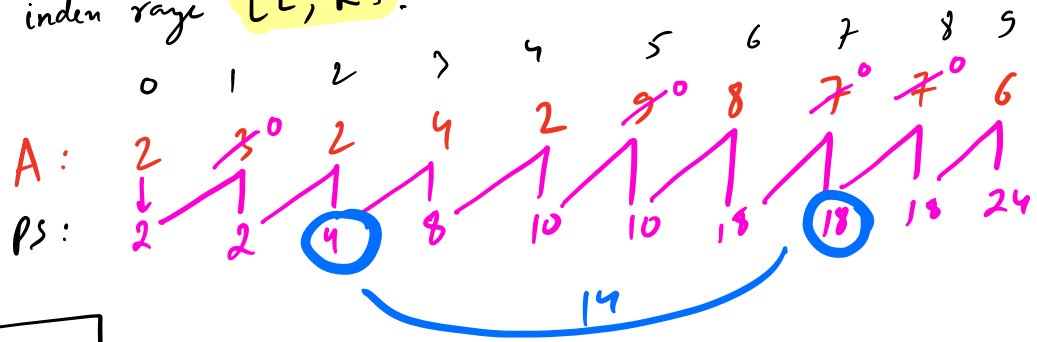
SC = O(N)

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
4	4	2
5	5	0

Idea: Assuming ODD no's are 0!

```
int 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);
}
```

N × O(1)

O(N)

```
for( i=1; i<=q; i++) {          ⟶ q      x O(1)
    // [L,R]
    if( L==0)
        sum = PS[R];                                  : O(q)
    else
        sum = PS[R] - PS[L-1];
    // sum ⟶ Ans of the ith query!
}
```

TC = O(N+q)

SC = O(N) → O(1)

---

int ⟶ -2×10^9, 2×10^9

long ⟶ -9×10^{18}, 9×10^{18}