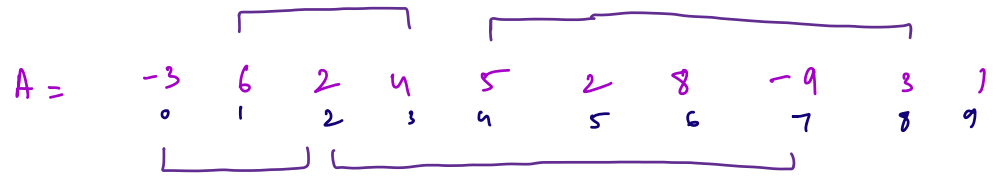
# Range Sum Queries

Given an array of size N and Q queries of the format start index (s) and end index (e) find sum of elements from index s to e




N = 10, Q = 4

| S | E | Sum |
|---|---|---|
| 1 | 3 | 6+2+4 = 12 |
| 2 | 7 | 12 |
| 4 | 8 | 9 |
| 0 | 2 | 5 |

Constraints

$N \leq 10^6$

$Q \leq 10^6$

$A[i] \leq 10^9$

int: $[-2 \times 10^9, 2 \times 10^9]$

long: $[-10^{18}, 10^{18}]$

Approach1: (Brute Force)

for(i=i; i<=Q; i++)

```
for(int i=0; i<Q; i++){
    scan(s,e)                    // Take s,e as input

    long
    int sum=0;
    for(int j=s; j<=e; j++){
        sum = sum + A[j];
    }
    print(sum);
}
```

Q

N

T.C: $O(Q \times N)$ ✓

S.C: $O(1)$

| Case1 | Case2: |
|---|---|
| s = 0 | s = 0 |
| e = N-1 | e = 0 |

#byte = 8·N°

$O(N^0) = O(1)$

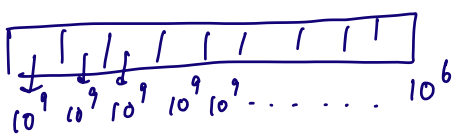
$N \leq 10^6$

$Q \leq 10^6$

$A[i] \leq 10^9$ ✓

#Iters = $10^6 \times 10^6$

= $10^{12}$ iters ⇒ TLE




sum = $10^9 + 10^9 + 10^9 + \cdots$ ($10^6$ times)

sum : $10^9 \times 10^6 = 10^{15}$

Approach2:

250-3, 45 overs

| Scores : | 2 | 8 | 14 | 29 | 31 | 49 | 65 | 79 | 88 | 97 |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1st | 2nd | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

Ex: Runs(10) = 97 − 88 = 9

(97: score at the end of 10th over; 88: score at the end of 9th over)

Quiz1: Scores :

| | 2 | 8 | 14 | 29 | 31 | 49 | 65 | 79 | 88 | 97 |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1st | 2nd | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

Runs(7) = Score[7] − Score[6]

= 65 − 49 = 16

Quiz2 :

| Scores : | 2 | 8 | 14 | 29 | 31 | 49 | 65 | 79 | 88 | 97 |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1st | 2nd | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

Runs(6,10) = Score[10] − score[6−1]

= 97 − 31 = **66**

Quiz3:

| Scores : | 2 | 8 | 14 | 29 | 31 | 49 | 65 | 79 | 88 | 97 |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1st | 2nd | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

Runs(3,6) = Score[6] - Score[3-1]

= 49 - 8 = 41

Runs(L, R) = Score[R] - Score[L-1]

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| A = | -3 | 6 | 2 | 4 | 5 | 2 | 8 | -9 | 3 | 1 |
| P.S = | -3 | 3 | 5 | 9 | 14 | 16 | 24 | 15 | 18 | 19 |

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[2] + A[3]

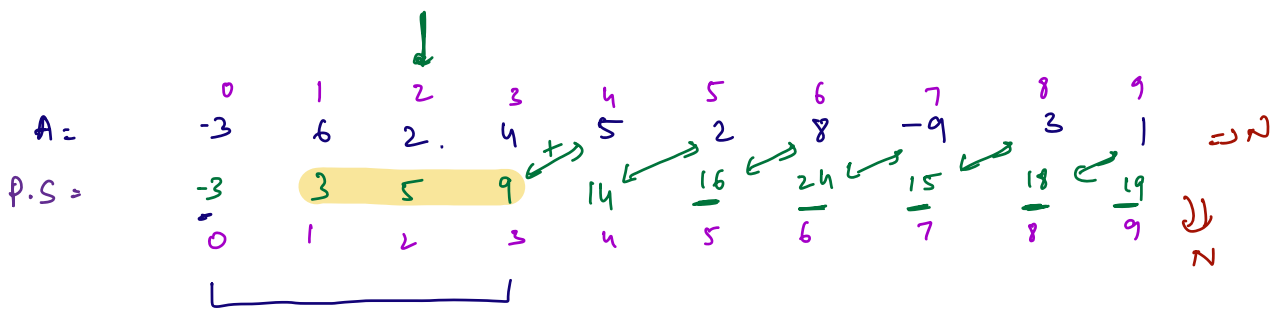
PS[4] = PS[3] + A[4]

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

sum(0....i) sum(0....i-1)

```
void computePrefixSum(int[] A, int N){
    long[N] PS;
    PS[0] = A[0];
    for(i = 1 ; i<N; i++){
        PS[i] = PS[i-1] + A[i];
    }
}
```

T.C: O(N)




| S | E | Sum |
|---|---|---|
| 1 | 3 | PS[3] − PS[1−1] = 9 − (−3) = 12 |
| 2 | 7 | PS[7] − PS[2−1] = 15 − 3 = 12 |
| 4 | 8 | PS[8] − PS[4−1] = 18 − 9 = 9 |
| 0 | 2 | PS[2] − PS[0−1] = 5 − = |

Sum(L ... R) = PS[R] − PS[L−1]

O(N) ← // Construct the Prefix Sum array

```
for( i=0; i<Q; i++){
    // Scan (s, e)
    if(s==0){
        print( PS[e] );
    }
    else{
        print( PS[e] - PS[s-1] );
    }
}
```

O(Q) { O(1) { O(1)

T.C: O(Q) + O(N) ⇒ O(N+Q)
↳ construct prefix sum array

S.C: O(N)
↳ Prefix Sum array

$N = 10^6$
$Q = 10^6$
$N+Q = 2*10^6$

A: 3 2 4 1
0 1 2 3

A = 3 5 9 1
0 1 2 3

$N = 10^6$
$A[i] = 10^9$

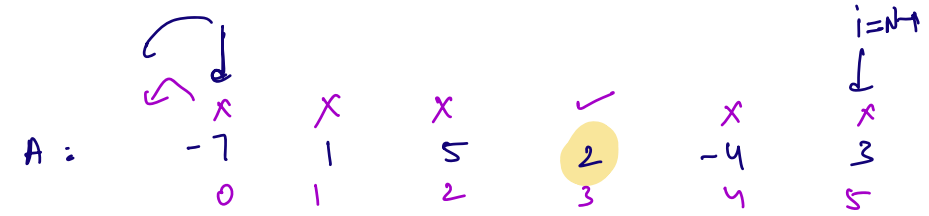
```
for(i=1; i<N; i++){
    A[i] = A[i-1] + A[i];
}
```

$N \leq 10^6$
$A[i] \leq 10^9$

**Question:** Equilibrium Index

Given an array, count no. of equilibrium indices

**Equilibrium Index:** An index of an array such that sum of elements to left is equal to sum of elements to right

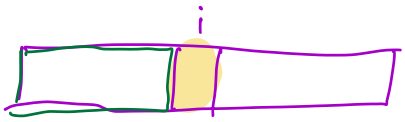



| | | |
|---|---|---|
| $i = 0$ | $S_L = 0$, | $S_R = 1+5+2-4+3 = 7$ |
| $i = 1$ | $S_L = -7$ | $S_R = 5+2-4+3 = 6$ |
| $i = 2$ | $S_L = -6$ | $S_R = 1$ |
| $i = 3$ | $S_L = -1$ | $S_R = -4+3 = -1$ |
| $i = 4$ | $S_L = 1$ | $S_R = 3$ |
| $i = 5$ | $S_L = -3$ | $S_R = 0$ |

ans = 1

8:41

Approach




$S_L = S_R$

$Sum(0, i-1) = Sum(i+1, N-1)$

```
int equilibrumIndices (int[] A, int N) {

O(N) ← //Construct prefix sum array [long[]]
       int count = 0;
       for (i=0; i<N; i++) {
            // SL = sum from 0 to i-1
            long SL;
            if (i == 0) {
                SL = 0;
            }
            else {
                SL = PS[i-1];
            }
            // SR = sum from i+1, N-1
            long SR = PS[N-1] - PS[i];

            if(SL == SR) {
                count++;
            }
       }
       return count;
}
```

N × O(1) = O(N)

N + N = 2N → O(N)

T.C: O(N) + O(N) = O(N)

S.C: O(N)
↳ Prefix sum array

$Sum(0, i-1) = PS[i-1]$

$Sum(L, R) = PS[R] - PS[L-1]$

$Sum(i+1, N-1) = PS[N-1] - PS[i+1-1]$

$= PS[N-1] - PS[i]$

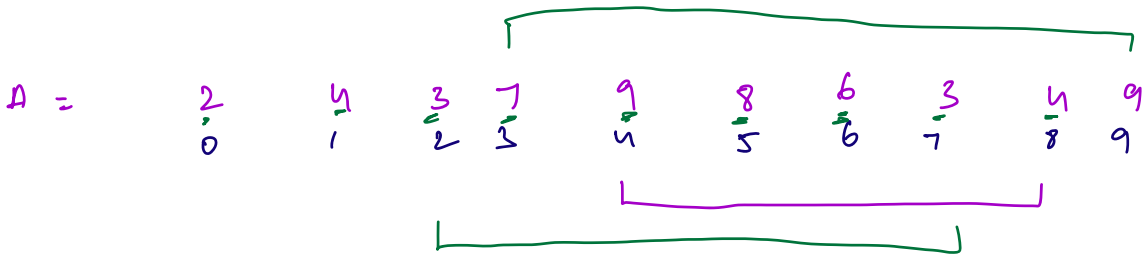
^

→ Sum of a range
→ product of a range
→ XOR of a range

} ⇒ Prefix Sum

**Question:** Count # even numbers in a range

Given an array and Q queries of format start (s) and end (e) index, find no. of even numbers in that range




Q = 5

N = 10

| s | e | ans |
|---|---|---|
| 4 | 8 | 1 + 1 + 1 = 3 |
| 3 | 9 | 3 |
| 2 | 7 | 1 + 1 = 2 |
| 0 | 4 | 2 |
| 0 | 5 | 3 |

$N \leq 10^6$
$Q \leq 10^6$
$A[i] \leq 10^9$

**Quiz:**

A = 5 2 3 4 16 20 7 12
0 1 2 3 4 5 6 7

Ans = 3

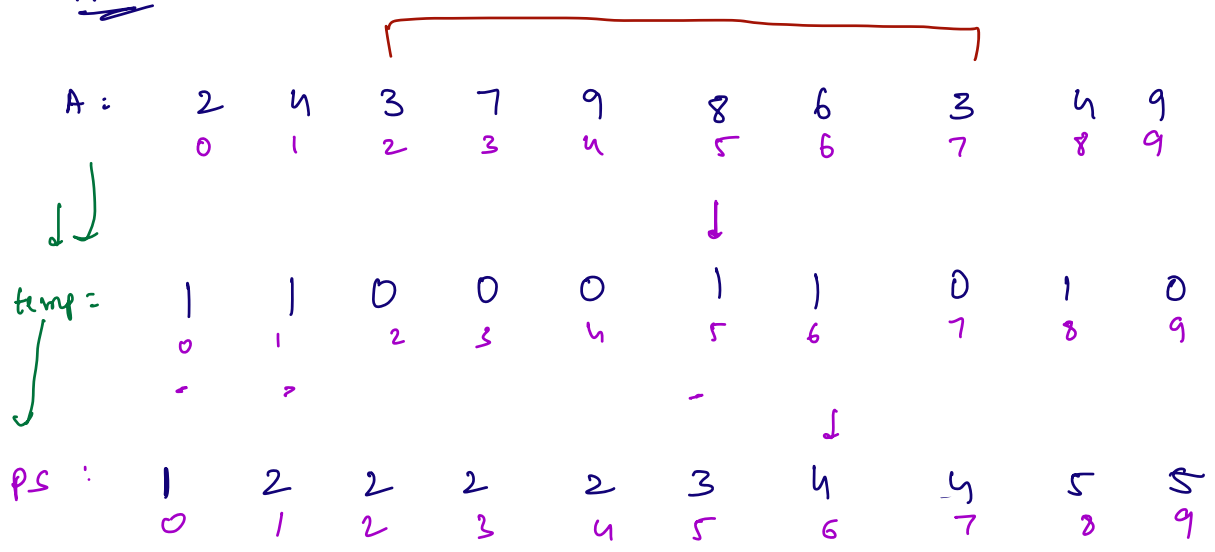
Brute Force

```
for (i=0; i<Q; i++) {
    // scan(s,e);
    int count = 0;
    for (j=s; j<=e; j++) {
        if (A[j] % 2 == 0) {
            count++;
        }
    }
    print(count);
}
```

O(N)

T.C: O(Q×N) -> TLE

Approach2:



No. of even [2, 7] => PS[7] − PS[2−1]
= 4 − 2 = 2

[1, 9] => PS[9] − PS[1−1]
= 5 − 1 = 4

[0, 5] => PS[5] = 3

```
void evenNumberInRange(int[] A, int N) {
    int[] PS;
    for(i=0; i<N; i++){
        if(A[i]%2==0) PS[i] = 1
        else PS[i] = 0
    }
    // Construct Prefix Sum
    for(i=1; i<N; i++){
        PS[i] = PS[i-1] + PS[i];
    }
    for(i=0; i<Q; i++){
        // scan (s, e)
        if(s==0){
            print(PS[e]);
        }
        else{
            print(PS[e] - PS[s-1]);
        }
    }
}
```

N

N

Q

T.C: $O(2N+Q) = O(N+Q)$

S.C: $O(N)$

↳ PS array

Following Metrics,

1) Problem Solving Percentage (PSP) ✓
2) Contest Performance =>
3) Mock Interviews => ✓
4) Attendance ✓

Weekday => 3 hrs

Saturday/Sunday => 8 hrs

=> Try for atleast 30-45 mins

=> Hints

=> TA

Weekend problem solving sessions

A: 5 4 1 2 3 6 7

0 1 2 3 4 5 6

[s, e)

A: 2 4 3 7 9 8 6 3 4 9

0 1 2 3 4 5 6 7 8 9