Array: Prefix Sum

Question 1

Given N array elements and Q queries on same array. for each query, calculate sum of all elements in a given range [L, R]. $0 <= L, R < n$ and $L <= R$

eg a[10] =

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

Q = 6

$0 <= L <= R < n$

0	4	8	= 5 + 2 + 8 + -9 + 3 = 9
1	3	7	= 10
2	1	3	= 12
3	0	4	
4	6	9	
5	7	7	= -9

let say, we have 2 arrays : L[Q], R[Q]

```
for( i=0; i<Q; ++i ) {
    //for each query, we get L[i], R[i]
    sum=0;
    for( j=L[i] ; j<=R[i] ; ++j ) {
        sum = sum + a[j];
```

→ max iterations = N

```
    }
    print(sum)
}
```

TC: O(8×N)

SC: O(1)

Given Indian team scores for first 10 overs of batting. After every over, current score is given.

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

1. Total runs scored in last over.

→ 97 - 88 = 9

2. Total runs scored in last 5 overs.

→ score[6-10]

score[1-10] = score[1-5] + score[6-10]

score[6-10] = score[1-10] - score[1-5]

97 - 31

= 66

3. Total run scored in 7th over ?

→ score[7-7] = 65 - 49

= 16

If we have a cumulative array, we can answer the range queries faster.

pf[n] pf[i] = a[0] + a[1] + ... + a[i]

pf[0] = a[0]

pf[1] = a[0] + a[1] ⇒ pf[0] + a[1]

pf[2] = a[0] + a[1] + a[2] ⇒ pf[1] + a[2]

pf[3] = a[0] + a[1] + a[2] + a[3] ⇒ pf[2] + a[3]

⋮

pf[i] = pf[i-1] + a[i]

How to create pf[] array?

```
pf[0] = a[0]
for (i=1; i<n; ++i) {
    pf[i] = pf[i-1] + a[i]
}
```

Coming back to Question 1

If we create pf array on array a[n]

	0	1	2	3	4	5	6	7	8	9
a[10] =	-3	6	2	4	5	2	8	-9	3	1
				↓ L				↓ R		

pf[R] = sum[0, R]

sum[L, R] = ?

sum[0, R] = sum[0, L-1] + sum[L, R]

sum[L, R] = sum[0, R] - sum[0, L-1]

= pf[R] - pf[L-1]

Code

```
pf[n]
pf[0] = a[0]
for (i=1; i<n; ++i) {
    pf[i] = pf[i-1] + a[i]
}                                  → N-1 iterations
3
for ( i=0; i<8; ++i) {             → 8 iterations
    // L[i], R[i] ⇒ left index, right index
    if ( L[i] == 0)
        sum = pf[R[i]]
    else
        sum = pf[R[i]] - pf[L[i]-1]
print (sum)
3
```

[0, 3] ⇓ pf[3]

if input is [0, R] ans = pf[R]

TC: O(N+8)

or

O(max(N, 8))

SC: O(N)

Instead of creating new pf[] array, we can modify existing array.

pf[0] = a[0]

pf[1] = a[0] + a[1]

pf[2] = a[0] + a[1] + a[2]

pf[0] a[0]

pf[1] a[1] = a[1] + a[0]

pf[2] a[2] = a[2] + a[1]

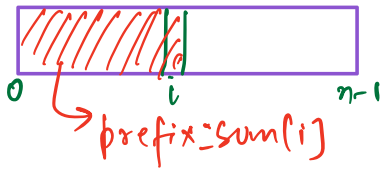
pf[i] a[i] = a[i] + a[i-1]

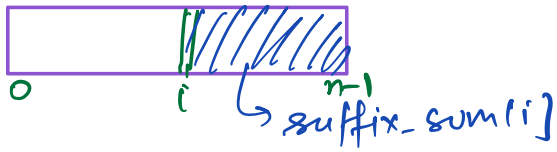
```
for (i=1; i<n; ++i ) {
    a[i] = a[i] + a[i-1].
}
```

This will reduce SC: O(1)

NOTE: This can only be done if you are allowed to modify the input array.

1. If you do sum from index 0 (from start)
   ⇒ sum(0.... i) = prefix_sum[i]
2. If yo do sum from index n-1 (from end)
   ⇒ sum[i.... n-1] = suffix_sum[i]





Question2 : Equilibrium Index

Given N array elements, count number of equilibrium index.

↳ index i is equi. index iff

left right

sum[0, i-1] = sum[i+1, n-1]

(before i) (after i)

if i = 0

left sum = 0

if i = n-1

right sum = 0

eg a[4]

	0	1	2	3
a[4]	-3	2	4	-1
left	0	-3	-1	3
right	5	3	-1	0

equal

count = 1

1. Create prefix sum array : pf[n]

↳ N iterations

```
count = 0
for (i=0; i<n; ++i) {   → N iterations
    if(i==0)
        left=0
```

```
else
    left = pf[i-1]    // sum[0, i-1]

if (i == n-1)
    right = 0
else
    right = pf[n-1] - pf[i]    // sum[i+1, n-1]

if (left == right)
    count++
}
print(count)
```

TC : O(N)

SC : O(N)
↳ pf array

Question 3

Given N array elements & Q queries.
For each query [L, R], find number of even numbers in the given range.

a[] : 2 4 3 7 9 8 6 5 4 9

b[] : 1 1 0 0 0 1 1 0 1 0

↳ same as sum of elements in a given range