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]. $0 <= L, R < n$ & $L <= R$

eg a[10] = (N=10)

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

Q = 6

$0 <= L <= R < n$

L[Q]	R[Q]	
4	8	= 5 + 2 + 8 + -9 + 3 = 9
3	7	= 4 + 5 + 2 + 8 + -9 = 10
1	3	= 12
0	4	
6	9	
7	7	= -9

Let say we have 2 arrays to represent queries:

L[Q] , R[Q]

```
void sum(int Q, int N, int a[], int L[], int R[]) {
    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 += a[j]
        }
        print(sum)
    }
}
```

→ max iterations = N

TC : O(Q×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 from 6th to 10th over. [6,10]

   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 runs scored in 7^{th} over ?

score [7-7] = score [1-7] − score [1-6]

= 65 − 49

= 16

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

a[n] ⟶ create a cumulative array pf[n] (pf → prefix sum)

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

pf[0] = a[0]

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

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

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

⇒ pf[1] = pf[0] + a[1]

⇒ pf[2] = pf[1] + a[2]

⇒ pf[3] = pf[2] + a[3]

⋮

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

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

→ O(N) : TC

O(N) : SC

If we create pf array on array a[n]

N=10

eg a[10] = -3 6 2 4 5 2 8 -9 3 1

0 1 2 3 4 5 6 7 8 9

L = 4, R = 8

pf[R] = sum[0,R]

sum[L,R] = ?

sum[4,8] = ?

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

```
for(i=0; i<Q; ++i) {
    // 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)
}
```

→ Q iterations

TC : O(N+Q)

SC : O(N)

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

pf[0] = a[0]

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

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

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

a[2] = a[2] + a[1] (pf[2], pf[1])

⋮

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

This will reduce SC : O(1)

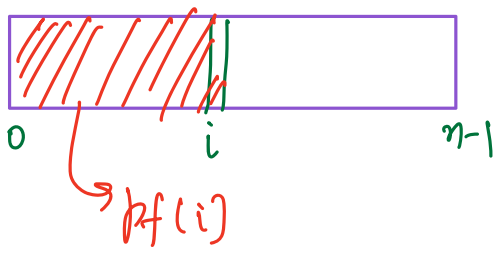
NOTE: This can only be done if you are allowed to modify the input array.

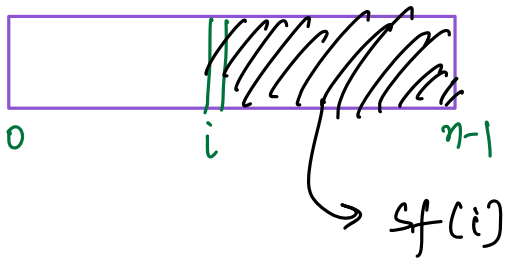
1. If you do sum from index 0 (from start)

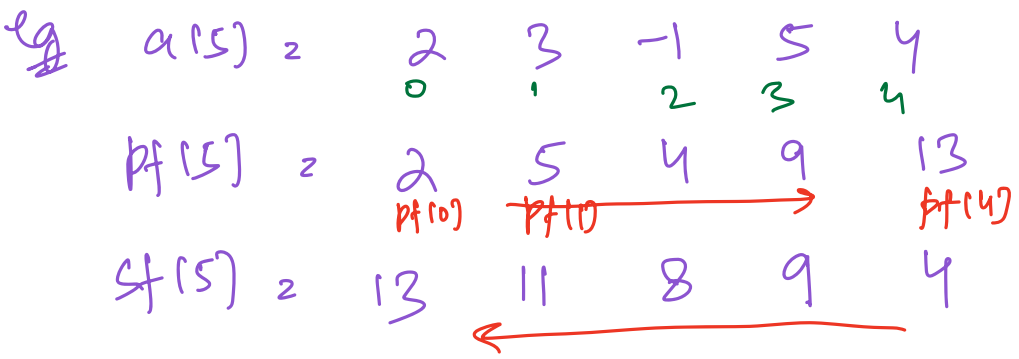
   ⇒ sum(0, 1, ..., i) = prefix_sum[i]

2. If you do sum from index n-1 (from end)

   ⇒ sum(i, i+1, ..., n-1) = suffix_sum[i]








```
sf[n-1] = a[n-1]
for (i = n-2; i >= 0; --i) {
    sf[i] = sf[i+1] + a[i]
}
```

BREAK : 8:24 - 8:34

Question 2 : Equilibrium Index

Given N array elements, count number of equilibrium index.

index i is equi. index iff.

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

(before i) (after i)

if i = 0
left sum = 0

if i = n-1
right sum = 0

eg a[4] = -3 2 4 -1

	0	1	2	3
left	0	-3	-1	3
right	5	3	-1	0

count = 1

1. Create prefix sum array : pf[N]

TODO
↳ O(N)

```
count=0

for(i=0; i<n; ++i) {
    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]
                ↳ sf[i+1]

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

(in `pf[n-1]`: n-1 → R; in `pf[i]`: i → L-1; in `sum[i+1, n-1]`: i+1 → L, n-1 → R)

TC : O(N)

SC : O(N)

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.