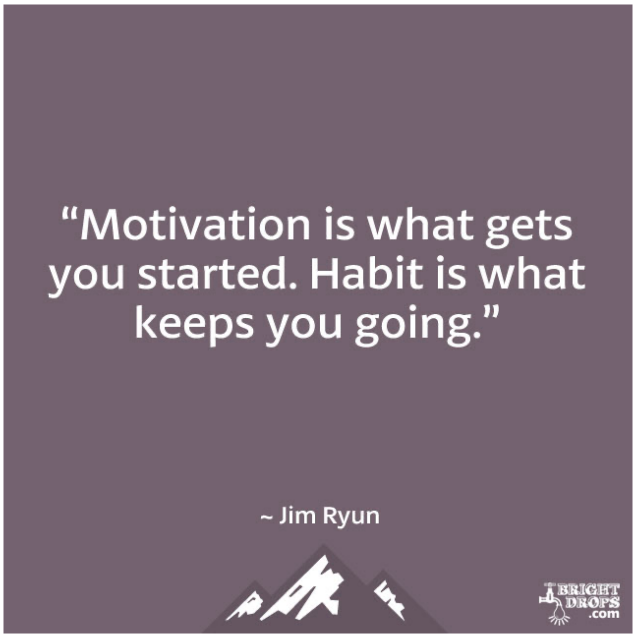
Prefix Sum



Good
Evening

Today's content

01. Range query
02. Prefix sum construction
03. Equilibrium Index
04. Even numbers in given range

01. Given ar[N] elements & Q queries

For each query : Given L & R. Calculate & print sum of all elements in range [L R]. L & R both has to be calculate

Note :- L & R are array indices such that $0 \le L \le R < N$

Constraints

$1 \le N, Q \le 10^5$
$-11 \le A[i] \le 10^9$
$0 \le L \le R < N$

$O(Q * N) \Rightarrow 10^5 * 10^5$
$\Rightarrow 10^{10} \rightarrow$ TLE

ar[10] =

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

Queries = 5

L	R	sum
4	8	→ 9
3	7	→ 10
1	3	→ 12
0	4	→ 14
7	7	→ -9

BF idea → For each query, iterate from L to R, get sum & print it.

```
void Rangesum ( int []ar, int Q ,int [] L,int []R)
    for ( i=0 ; i<Q ;i++ )
        int s=L[i], e=R[i], long sum=0

        for ( j=s; j≤e ; j++){
            sum= sum+ ar[j];
        }
        print (sum):
    }
}
```

TC=O(Q*N)
SC=O(1)

Q: Given Indian Cricket Team score, for first 10 overs of Batting.

After every over, total score is given as:

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

01. Total no. of runs scored in 10^{th} over

total runs after 9^{th} over + runs scored in 10^{th} over = Total runs scored after 10^{th} over

$$88 + x = 97$$

$$x = 97 - 88 \Rightarrow 9$$

02. Runs scored in 7^{th} over

$$sc[7] - sc[6] = 65 - 49$$

$$= 16$$

03. Runs scored from 6^{th} to 10^{th} over = $[6-10]$

$$sc[1-10] = sc[1-5^{th}] + sc[6-10]$$

$$97 = 31 + x$$

$$x = 97 - 31$$
$$= 66$$

04. Runs scored from 3^{rd} over to 6^{th} over

$sc[6] \rightarrow$ total runs from $[1-6]$

$sc[2] \rightarrow$ total runs from $[1-2]$

$$\underbrace{[1-6]}_{49} = \underbrace{[1-2]}_{8} + [3-6]$$

$$49 = 8 + x$$

$$x = 49 - 8 = \underline{\underline{41}}$$

Q5. Runs scored from 1^{st} to 5^{th} over = sc[5]

= 31

ar[10] =

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

Pf [10] =

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

Pf [3] → sum of all ele from [0-3]

Pf [5] → sum of all ele from [0-5]

Q1. sum[4 - 8]

$$sum[0-8] = \underbrace{sum[0-3]} + sum[4-8]$$

$$Pf[8] = pf[3] + x$$

$$x = pf[8] - pf[3]$$

$$= 18 - 9 = 9$$

02. sum[3 7]

$$\underbrace{sum[0-7]}_{pf[7]} = \underbrace{sum[0-2]}_{pf[2]} + \underbrace{sum[3-7]}_{req\ sum}$$

$$req\ sum = pf[7] - pf[2]$$
$$= 15 - 5$$
$$= 10$$

03. sum[1 - 3]

$$\underbrace{sum[0\ 3]}_{pf[3]} = \underbrace{sum[0\ 0]}_{pf[0]} + \underbrace{sum[1\ 3]}_{req.\ sum}$$

$$req\ sum = 9 - (-3)$$
$$= 12$$

04. sum[7 7]

$$sum[0\ 7] = sum[0\ 6] + sum[7\ 7]$$
$$sum[0\ 7] - sum[0\ 6] = sum[7\ 7]$$
$$15 - (-24) = sum(7\ 7)$$

sum[i j]

$$sum[0\ j] = sum[0\ \ i-1] + sum[i\ j]$$

$$\underbrace{}_{pf[j]} = \underbrace{}_{pf[i-1]} + \text{req sum}$$

$$sum[i\ j] = pf[j] - pf[i-1]$$

$$\text{Q1}\quad sum[0\ 4] = pf[4] - pf[0-1]$$

$$= pf[4] - \underbrace{pf[-1]}_{\text{invalid index}}$$

Generic formula
$$\begin{cases} i == 0 & : \ pf[j] \\ i\,!= 0 & : \ pf[j] - pf[i-1] \end{cases}$$

Construction of pf array

$ar[10] =$

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

$Pf[10] =$

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

$$Pf[0] = ar[0]$$

$$Pf[1] = \underbrace{ar[0]} + ar[1]$$

$$Pf[0] + ar[1]$$

$$Pf[2] = \underbrace{ar[0] + ar[1]} + ar[2]$$

$$Pf[1] + ar[2]$$

$$Pf[3] = \underbrace{ar[0] + ar[1] + ar[2]} + ar[3]$$

$$Pf[2] + ar[3]$$

$$Pf[i] = \underbrace{ar[0] + ar[1] + \dots + ar[i-1]} + ar[i]$$

$$Pf[i-1] + ar[i]$$

Generalised expression

$$Pf[i] = Pf[i-1] + ar[i]$$

$$Pf[0] = Pf[0-1] + ar[0]$$

Error

Code

```
Pf[0] = ar[0]

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

→ Construction

⇒ Optimise code for first question.

```
void Rangesum ( int[n] ar, int Q, int[]L, in[]R)
    // Construct pf array

        long [] pf = new long [n]
         pf[0] = ar[0].
         for (i=1; i<n; i++)
             pf[i] = pf[i-1] + ar[i];
         }
```

O(n)

// Answer all queries

```
for ( i=0; i<Q ; i++)
    int s = L[i] , e = R[i]
    if (s == 0)  print (pf[e])
    else  print (pf[e] - pf[s-1]);
}
}
```

$O(Q)$

$TC = O(N+q)$

$SC = O(N)$

Can Optimise this space complexity to O(1) by modifying the given arr

Modify the given arr

arr[10] =

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

Pf [10] =

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

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

Dry run

pf = | 3 | 9 | 11 | 12 |

	9	11	
3	~~6~~	~~2~~	~~1~~ 12
0	1	2	3

10:12 → 10:22 pm

i = 1	$1 < 4$
~~2~~	$2 < 4$
~~3~~	$3 < 4$
4	$\boxed{4 \leq 4}$ → break

$$ar[3] = ar[3-1] + ar[3]$$
$$= ar[2] + ar[3]$$
$$= 11 + 1$$
$$= 12$$

02. Equilibrium Index

Given N array elements, count no. of equilibrium idx

An idx i is known as eqm idx

sum of all elements before i idx == sum of all elements after i idx

0 1 2 3 . . . i-1	i	i+1 i+2 . . . n-1
leftsum [0 i-1]	==	rsum [i+1 n-1]

Note :- if (i == 0) left sum = 0

if (i == N-1) righsum = 0

	0	1	2	3
ar [4] =	-3	2	4	-1
leftsum =	0	-3	-1	3
rightsum =	5	3	-1	0

Ans = 1

	0	1	2	3	4	5	6
ar [7] =	-7	1	5	2	-4	3	0
left sum =	0	-7	-6	-1	1	-3	0
right sum	7	6	1	-1	3	0	0
	✗	✗	✗	✓	✗	✗	✓

Ans = 2

Brute force → For each & every index,
calculate the lsum & rsum.
if (lsum == rsum) count = count + 1.

```
int equilibrium ( int [n] ar )
    count = 0

    for ( i = 0 ; i < n ; i++ )
        int lsum = 0            // sum → [0   i-1]

        for ( j = 0 ; j ≤ i-1 ; j++ ) {
            lsum += ar [j] ;
        }

        int rsum = 0        // sum [i+1 → n-1]

        for ( j = i+1 ; j < n ; j++ )
            rsum += arr [j] ;
        }

        if (lsum == rsum) count = count+1;
    }
    return count;
}
```

$TC = O(n^2)$

$SC = O(1)$

Optimised soln

For a particular index i

└ lsum = [0 i-1]

└ rsum = [i+1 n-1]

can use psum array to get these answer

```
int equilibrium ( int [] arr)
    → Construct pf[n] ———→ {TODO}
    for (i=0; i<n; i++)
        int lsum=0            //lsum → [0 i-1]
        if(i>0) { lsum= pf [i-1]; }
        int rsum=0;           //rsum → [i+1 n-1]
          rsum= pf[n-1] - pf[i]
        if (lsum = = rsum) count= count +1;
    }
    return count;
}
```

TC = O(n)

SC = O(n)

Q3. Given ar[N] elements & Q queries.

For each query : Given L & R calculate & print no. of even numbers in given range [L R]

num % 2 == 0

Note :- L & R are array indices such that $0 \le L \le R < N$

Constraints

$1 \le N, Q \le 10^5$

$1 \le A[i] \le 10^9$

$0 \le L \le R < N$

Eg:- ar[10] :

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

Queries : 4

L[4]	R[4]	
4	8	→ 3
3	9	→ 3
2	7	→ 2
0	4	→ 2

Brute force → For each query iterate from L to R & calculate the no of even elements

```
void Rangesum ( int [] ar, int Q , int [] L, in[] R)
{
    for ( i=0 ; i<Q ; i++ )
    {
        int s = L[i] , e = R[i] , int count = 0
        for ( j=s ; j≤e ; j++ ){
            if ( ar[j] % 2 == 0) { count = count +1 }
        }
        print (count);
    }
}
```

$TC = O(Q * N)$

$SC = O(1)$

At max, Q & $N = 10^5$

$TC = O(10^5 * 10^5) = \boxed{10^{10}}$ TLE ✗

Optimisation

Prefix array that has the count of even no.

arr =

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

pf =

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

Obs

01. When curr ele == even → +1

02. If curr ele == odd → +0

Pf array

```
if ( arr[0] % 2 == 0) pf[0] = 1
else pf[0] = 0
for (i=1 ; i<n ; i++ ){
    if ( arr[i] % 2 == 0) pf[i] = pf[i-1]+1 ;
    else pf[i] = pf[i-1] +0 ;
}
```

Queries : 4

L[4]	R[4]		
4	8	→ pf[8] − pf[3]	→ 3
3	9	→ pf[9] − pf[2]	→ 3
2	7	→ pf[7] − pf[1]	→ 2
0	4	→ pf[4]	→ 2

Idea 2 → psum array

arr =

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

pfsum = 1 2 2 2 2 3 4 4 5 5

```
void Rangesum ( int [] ar, int Q , int [] L, int [] R)
    construct pf array ⟶ {TODO}
    for ( i=0 ; i<Q ; i++ )
        int s = L[i] , e = R[i] , int count = 0
        if (s == 0) print ( pf [e] ;
        else print ( pf [e] - pf [s-1] ;
    }
}
```

Prefix Array → Whenever you have range queries

→ Whenever you have to work on continous part of an array

Doubts

find max ele & max ele freq

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

```
int max = ar[0]

int count = 1

for (i=1 ; i<n ; i++)

    if (ar[i] == max){
        count = count+1
    }
    else if (ar[i] > max){
        max = arr[i]
        count = 1
    }
}
```