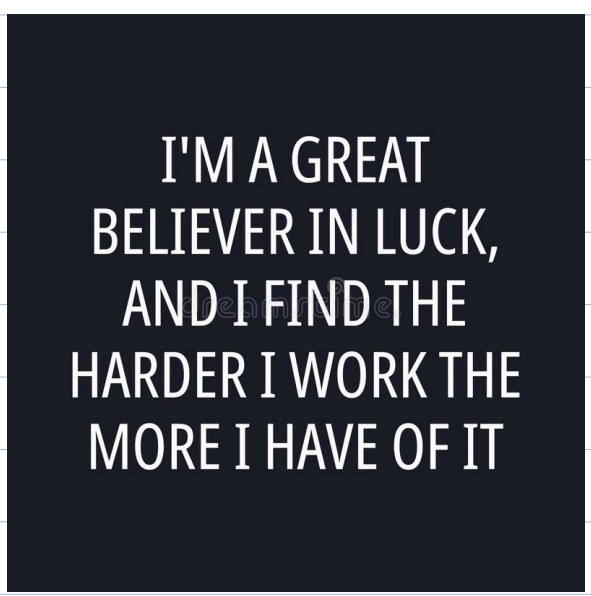
Today's Content

→ Prefix Sum

→ Problems on PSum.

→ Todo :- Doubt Session today.



Ques Given an array of size n, and Q queries of the format s and e.

Return sum of elements from index s to e.

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

Q : 4

s	e	
1	3	: 12
2	7	: 12
4	8	: 9
0	2	: 5

T.C → O(q*n)

S.C → O(1)

Basic Idea

```
for(j=1; j<=q; j++){
    sum=0;
    Read s & e
    for(i=s; i<=e; i++){
        sum += arr[i];
    }
    Print(sum);
}
```

Given with total score after every over of last 10 over:

41	42	43	44	45	46	47	48	49	50
288	312	330	349	360	383	394	406	436	439

Cummulative data

Runs scored in last 5 over:-
[46-50]

Runs [50] − Runs [45] = 439 − 360 = 79

Last Over 50th over : 3 runs;

Runs in 49th over = R(49) − R(48)
= 30

Runs in 42nd to 45th over,
360 − 288 = 72

Prefix Sum:- Sum of all elements from 0 to ith idx.

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

pf contains sum of elements from 0 to i.

pf[5] = pf[4] + arr[5]

[0-5] [0-4]

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

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

pf[0] = pf[-1] + A[0]

if i == 0;

pf[0] = A[0];

```
if i==0;
    pf[i]= A[i];
else;
    pf[i]= pf[i-1] + A[i]
```

Pseudo code for Prefix Array:-

```
int pf[N];
for(i=0; i<N; i++){
    if(i==0){
        pf[i]= A[i];
    }else{
        pf[i]= pf[i-1] + A[i];
    }
}
```

T.C → O(n)

S.C → O(n)

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

s	e		
1	3	→	pf[3] - pf[0] => 9 - (-3) = 12
2	7	→	pf[7] - pf[1] => 15 - 3 = 12
4	8	→	pf[8] - pf[3] = 18 - 9 => 9
0	2	→	pf[2] = 5

s e

```
if (s==0) {
    sum = pf[e]
} else {
    sum = pf[e] - pf[s-1]
}
```

n → // prefix array created.

q →

```
for (j=1; j<=q; j++) {
    if (s==0) {
        sum = pf[e]
    } else {
        sum = pf[e] - pf[s-1]
    }
}
```

T.C →
O(n+q)

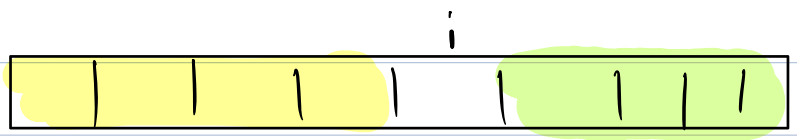
S.C →
O(n)

Ques) Given an array of size N, find Equilibrium index.

↓

index is called eq. idx.

Sum of all elements to its left = Sum of all elements to its right.



e.g.1)

0	1	2	3	4	5
1	2	3	4	8	10

Ans = 4.

e.g.2)

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

T.C → O(N), S.C → O(N).

// Create a pf first → O(n)

O(n) → for (i=0; i<n; i++) {

    // checking ith idx is eql idx or not

    int sumleft = pf[i-1];

    int sumright = pf[N-1] - pf[i]

    if (sumleft == sumright) {

        print(i);

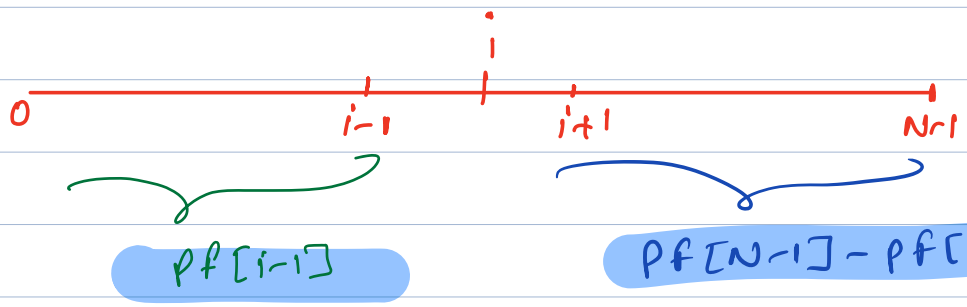
    }

}

Edge case for i=0. ↓ Todo.

s e

pf[e] - pf[s-1]



Break 10:06 - 10:16 pm.

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

-3

we can create pf array within same array itself.

// pseudo code to create pf array in the orig array.

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

Ques) Given N array elements and Q queries.
For each query l to r, find count of even numbers in the range.

	0	1	2	3	4	5	6	7	8	9
Eg:- arr[10] : {	2,	4,	3,	7,	9,	8,	6,	5,	4,	9}

Queries 3

l	r
4	8

3

// Pseudo Code

Brute force:-

```
for(j=1; j<=q; j++)
    read l to r
```

3 9 3

0 4 2

T.C -> O(Q*n)

S.C -> O(1)

```
count=0;
for(i=l; i<=r; i++){
    if(arr[i] %2==0){
        count++
    }
}
print(count);
```

3

3

Optimization :-

arr[10] : { 2, 4, 3, 7, 9, 8, 6, 5, 4, 9 }
0 1 2 3 4 5 6 7 8 9

even odd

arr[10] : { 1, 1, 0, 0, 0, 1, 1, 0, 1, 0 }
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

l r

3 5

0 6

pf[r] - pf[l-1]

3 - 2 => 1

Optimization

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

for (i=1; i<n; i++) {                    → O(N)
    arr[i] = arr[i-1] + arr[i]
}
```

q - input.

O(q) →

```
for (j=1; j<=q; j++) {
    l, r → input.
    if (l == 0) { print (arr[r])
    else { arr[r] - arr[l-1]);
}
```

T.C → O(N+q)

S.C → O(1).

Special Index

	0	1	2	3	4	5
	4	3	2	7	6	-2
pfodd →	0	3	3	10	10	8
pfeven →	4	4	6	6	12	12

→

0	1	2	3	4
4	3	7	6	-2

Sodd = 9 Seven = 9

Prefix Odd

Prefix Even

arr[7] = { -3, -2, 6, 8, 4, 8, 5 3

0 1 2 3 4 5 6

max

secondMax

max = ~~-3 -2 6~~ 8

freq = ~~x x x~~ x 2