Today's Content } → 2 classes

→ Prefix Sum Introduction

→ Problems based on Prefix Sum

Quote

I wanted the reward and not the struggle.
I wanted the result and not the process.
I was not in love with the fight but only the victory.
And life doesn't work that way.

Ques Given N array elements & Q queries in same array.

for each query calculate sum of all elements in Given range.

	0	1	2	3	4	5	6	7	8	9
arr[10]=	-3	6	2	4	5	2	8	-9	3	1

Q=6

L	R	
4	8	: 9
3	7	: 10
1	3	: 12
0	4	: 14
6	9	: 3
7	7	: -9

idea :-

for all queries, iterate over L-R & get sum

Pseudo code

```
// Given arr[N]
for (j=0; j<Q; j++) {
    Read L & R
    int sum=0
    for (i=L; i<=R; i++) {
        sum= sum+arr[i]
    }
    Print (sum)
}
```

T.C → O(Q*N)

S.C → O(1) → Constant

// Given Indian Team Scores for first 10 overs of Batting After every current Score given.

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

→ cumulative sum

① Total no. of runs scored in last overs → 97-88 = 9

② Total runs scored in last 5 overs = 97-31 = 66

③ Total runs made in 7th over = 65-49

④ Total runs made from [3-6] over ⇒ 49-8

⑤ Total runs scored from [1-5] over → 31

⑥ Total runs scored from [4-8] over ⇒ 79-14

	0	1	2	3	4	5	6	7	8	9
arr[10] =	-3	6	2	4	5	2	8	-9	3	1
pf[10] =	-3	3	5	9	14	16	24	15	18	19

pf ↳ prefix array

pf[3] = 9 => Sum of all element from [0 - 3]

pf[5] = 16 => Sum of all elements from [0 - 5]

Sum[$\overset{L}{4}$ - $\overset{R}{8}$] = arr[4] + arr[5] + arr[6] + arr[7] + arr[8]

Add Sum [0 3] on both sides

→ Sum[0 3] + Sum[4 - 8] =

arr[4] + arr[5] + arr[6] + arr[7] + arr[8]
+ arr[0] + arr[1] + arr[2] + arr[3]

= pf[8]

→ Sum[0 3] + Sum[4 8] = pf[8]

→ pf[3] + Sum[4 8] = pf[8]

→ Sum[4 8] = pf[8] - pf[3]

* $sum[3 - 7] = pf[7] - pf[2]$

* $sum[5-9] = \underbrace{sum[0-9]} - sum[0-4]$

$\downarrow \qquad \downarrow$

$sum[5-9] = pf[9] \qquad pf[4]$

* $sum[2-6] = \underbrace{sum[0-6]} - \underbrace{sum[0\ 1]}$

$\downarrow \qquad \downarrow \qquad \downarrow$

$sum[2-6] = pf[6] - pf[1]$

$l <= R$

* $sum[L-R] = \underbrace{sum[0-R]} - \underbrace{sum[0-L-1]}$

$\downarrow \qquad \downarrow$

→ $sum[L-R] = pf[R] - pf[L-1]$

$sum[0-2] = pf[2] - \underbrace{pf[-1]}$

$\Downarrow$

$pf[2]$

$\downarrow$

array out of bound exception

$L-R \Rightarrow sum[L-R] \Rightarrow \begin{cases} \text{if } (L=0) : pf[R] \\ \text{else} : pf[R]-pf[l-1] \end{cases}$

sum[s-5] = pf[5] - pf[4].

Given arr[N] → pf[N]

pf[0] = arr[0]

pf[1] = arr[0] + arr[1]

$pf[2] = \underbrace{arr[0] + arr[1]} + arr[2]$

↓

pf[2] = pf[1] + arr[2]

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

pf[i] = pf[i-1] + arr[i].

$pf[0] = \underbrace{pf[-1] +} arr[0].$ → out of bond.

→ given arr[N].

```
int pf[N]
pf[0] = arr[0].
 for (i=1; i<N; i++) {
 |  pf[i] = pf[i-1] + arr[i]
 }
```

	0	1	2	3	4
arr =	5,	4,	3,	2,	10
pf	5	9	12	14	24

i = ~~0~~, ~~1~~ ~~2~~ 3

// Sum of Q Queries

// Given arr[N]

```
given arr[N].
    int pf[N]
    pf[0] = arr[0].
    for (i=1; i<N; i++) {
    |   pf[i] = pf[i-1] + arr[i]
    }
Read Q. → L & R
for (i=0; i<Q; i++) {
|     if(L==0) { Print(pf[R])
|     else {
|         Print(pf[R]-pf[L-1])
|     }
}
```

T.C → $O(Q+N)$

S.C → $O(\underline{N})$

10:30 → 10:40 pm

Ques) Equilibrium Index

Given N array elements, count no. of equilibrium Index ?

An index i is said to be equilibrium index if ?

Sum of all elements Before i^{th} index = Sum of all elements after i^{th} index

Sum [0 i-1] Sum [i+1, N-1]

Note :- if i==0, left=0, if i== N-1, right=0

	0	1	2	3
Ex:- arr[4] =	-3	2	4	-1
left =	0	-3	-1	3
right =	5	3	-1	0

ans = 1

Ex 2 :-

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

ans = 2

Ex 3 :-

		0	1	2
arr []	=	3	-2	2
Left	=	0	3	1
Right	=	0	2	0

Ans = 1

idea :- check if every index is equilibrium or not

```
int countEquilibrium(int arr[]) {
    int n = arr.length;
    int c = 0;
    int pf [N]; { Initialize value Todo }
    for (i = 0; i < N; i++) {
        // if index i is equilibrium index
        // left = sum[0 i-1] = pf[i-1]
        int left = 0
        if (i > 0) { left = pf[i-1] }
        // right = sum[i+1 N-1] => pf[N-1]- pf[i]
         int right = pf[N-1]- pf[i]
          if (left == right) { c = c+1 }
        }

     return c
}
```

T.C → O(N + N) = O(N)

S.C → O(N)

19:05 pm → revise it