

Fb (3)

Q1. Given N array ele & Q queries

For each query, calculate sum of ele in range from

L to R.

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

<u>L</u>	<u>R</u>	<u>sum</u>	<u>$L \leq R$</u>	<u>Brute Force</u>
4	8	9	for (i=1 ; i<=Q ; i++) {	
2	7	12	scan(L, R)	
1	3	12	sum = 0	
0	4	14	for (j=L ; j<=R ; j++) {	
7	7	-9	sum += arr[j]	
			}	
			print(sum)	
			}	

$O(Q \times N)$ TC
 $\approx O(1)$

Cricket scores.

Build up for
calculating
prefix
sum.

1	2	3	4	5	6	7	8	9	10	sum
score	2	8	14	29	31	49	65	79	88	<u>97</u>

Runs scored in the 10th over ? $97 - 88 = 9$

$$[10 \quad 10] \quad score[10] - score[9]$$

Runs scored in last 5 overs ? $97 - 31 = 66$

$$[6 \quad 10] \quad score[10] - score[5]$$

Runs scored in just 7th over $65 - 49 = 16$

$$[7 \quad 7] \quad score[7] - score[6]$$

$[3 \quad 6]$ over $\rightarrow 49 - 8 = 41$

$$[3 \quad 6] \quad score[6] - score[2]$$

$[4 \quad 9]$ $\rightarrow 88 - 14 = 74$

$$[4 \quad 9] \quad score[9] - score[3]$$

$A[0]$	0	1	2	3	4	5	6	7	8	9
	-3	6	2	4	5	2	8	-9	3	1
$Pf[0]$	-3	3	5	9	14	16	24	15	18	19

$$Pf[0] = A[0]$$

$$Pf[1] = A[0] + A[1]$$

$$Pf[2] = A[0] + A[1] + A[2]$$

$$Pf[3] = A[0] + A[1] + A[2] + A[3]$$

$$\vdots Pf[i-1] = A[0] + A[1] + A[2] + \dots + A[i-1]$$

$$Pf[i] = \underbrace{A[0] + A[1] + A[2] + \dots + A[i-1]}_{Pf[i-1]} + A[i]$$

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

$Pf[N]$

for(i=0; $i < N$; $i++$) {

$$| \quad Pf[i] = \underbrace{Pf[i-1]}_{\text{ }} + A[i];$$

$$| \quad Pf[0-1] = Pf[-1]$$

→ Array index out of bounds.

$\text{pf}[N]$ ↘
 $\text{pf}[0] = A[0]$
 for($i=1$; $i < N$; $i++$) {
 | $\text{pf}[i] = \text{pf}[i-1] + A[i]$;
 | }
 ↗ }
i : [1 N-1]
Iterations : N-1
 $\text{TC} \rightarrow O(N)$
 $\text{SC} \rightarrow O(1)$

$A[10]:$ 0 1 2 3 4 5 6 7 8 9
 -3 6 2 4 5 2 8 -9 3 1
 $\text{pf}[10] : -3 \ 3 \ 5 \ 9 \ 14 \ 16 \ 24 \ 15 \ 18 \ 19$

Queries

$$L \quad R \quad \text{sum}$$

$$4 \quad 8 \quad \text{pf}[8] - \text{pf}[3]$$

$$18 - 9 = 9$$

$$3 \quad 7 \quad \text{pf}[7] - \text{pf}[2]$$

$$15 - 5 = 10$$

$$i \quad j \quad \text{pf}[j] - \text{pf}[i-1]$$

$$0 \quad 4 \quad \text{pf}[4] - \underbrace{\text{pf}[-1]}_{\text{pf}[4]}$$

Optimised

① Create the prefix array. } $\begin{cases} TC \rightarrow O(N) \\ SC \rightarrow O(N) \end{cases}$]

② for ($i=1; i \leq Q; i++$) {

 Read L, R

 if ($L == 0$) ans = pf[R]

 else {

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

 }

 print (ans)

$\begin{cases} TC \rightarrow O(Q) \\ SC \rightarrow O(1) \end{cases}$

} $\begin{cases} 1 \text{ query} \rightarrow O(1) \\ Q \text{ queries} \rightarrow O(Q) \end{cases}$

$$TC \rightarrow O(N) + O(Q) \rightarrow O(Q+N)$$

$$SC \rightarrow O(N) + O(1) \rightarrow O(N)$$

Q2. Given N array elements, count no. of equilibrium index.

Goldman
Sachs
(14)

Fb (12)

Apple

expedia

Amaron

An index is said to be equilibrium index if \rightarrow

sum of all elements before ith index

$\stackrel{(1)}{=}$ sum of all elements after ith index

$[0 \dots i-1]$

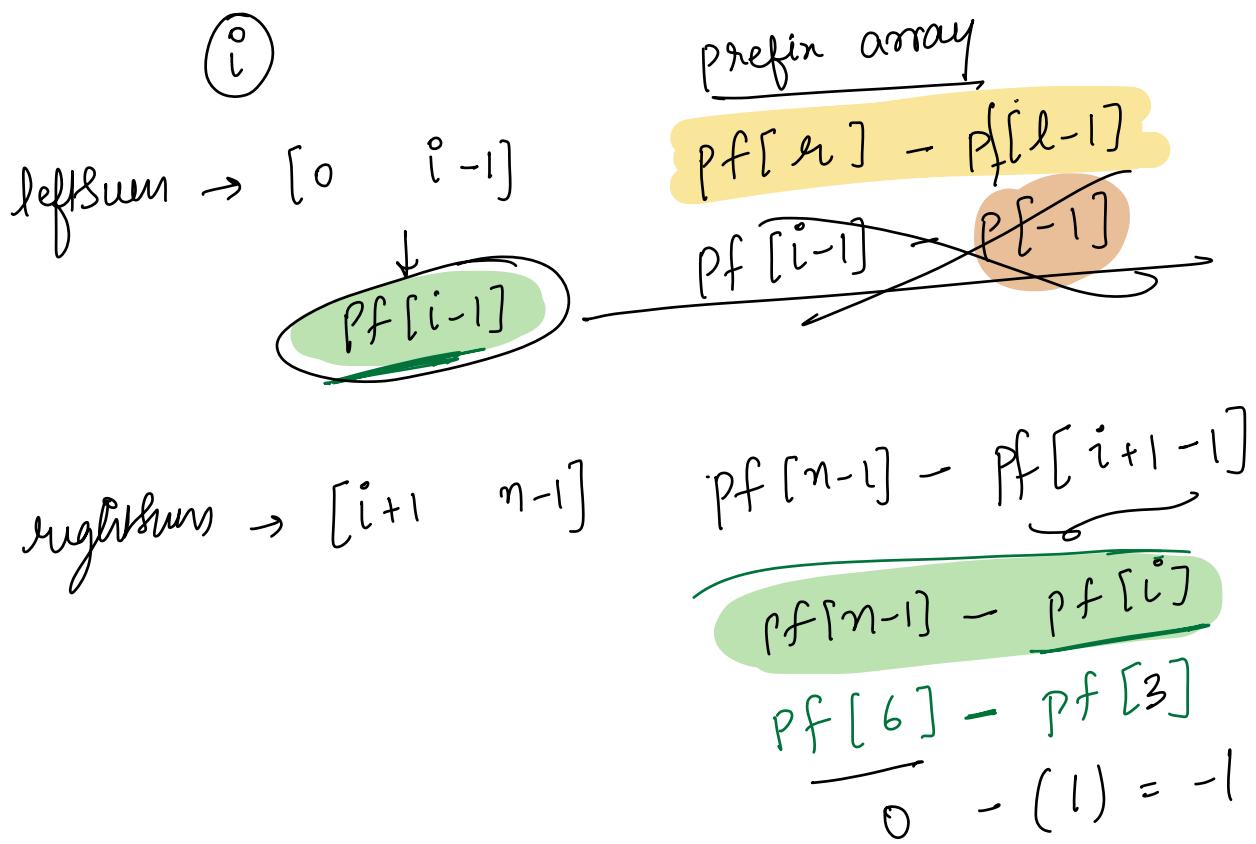
$[i+1 \dots n-1]$

	x	x	x		x	x	
ar[] =	0	1	2	3	4	5	
	-7	1	5	2	-4	3	
leftsum	0	-7	-6	-1	1	-3	0
rightsum	7	6	1	-1	3	0	0

ans: 2

	x	x	x		x	x	
ar[] =	0	1	2	3	4	5	
	3	-1	2	-1	1	2	1
leftsum	0	3	2	4	3	4	6
rightsum	4	5	3	4	3	1	0

ans: 2



	0	1	2	3	4	5	6
prefix array	-7	1	5	2	-4	3	0
leftsum	-7	-6	-1	1	-3	0	0
rightsum	6	1	-1				

Edge cases

~~(Ans)~~

$$i = 0$$

leftsum $\rightarrow \text{pf}[i-1]$

leftsum = 0

for
 $i = 0$

rightsum $\rightarrow \text{pf}[n-1] - \text{pf}[i]$

Range
Sum
Queries.

$\rightarrow \text{pf}[6] - \text{pf}[0]$

$i = n-1$

$n=7$

{

leftsum $\rightarrow \text{pf}[i-1]$
 $\text{pf}[n-2]$

$\text{pf}[5]$

rightsum $\rightarrow \text{pf}[n-1] - \text{pf}[i]$

$\text{pf}[6] - \text{pf}[5]$

① prefix sum array } TC $\rightarrow O(N) \leftarrow$
 SC $\rightarrow O(N)$

② int count = 0

```
for (i=0; i<N; i++) {
    leftsum = (i==0 ? 0 : pf[i-1])
    rightsum = pf[n-1] - pf[i]
    if (leftsum == rightsum) {
        count++
    }
}
```

return count.

TC $\rightarrow O(N) + O(N)$
 $= O(N)$
 $\underline{SC = O(N)}$

$$\text{sum} \rightarrow \frac{l \quad n}{pf[l] - pf[l-1]}$$

$[i+1 \quad n-1]$	$pf[n-1] - pf[i+1-1]$ $pf[n-1] - pf[i]$
-------------------	--

Break : 10:45

Qn 3. Given N array ele & Q queries,
for each query, calc sum of all even
indices in given range.

$$arr[] = \{ 3, 4, -2, 8, 6, 2, 1, 3 \}$$

8 queries

L	R	sum
2	5	4
3	7	7
0	7	8
4	4	6
5	5	0

Do u even need to look at
odd indices? — NO.

$$arr[] = \{ 3, 4, -2, 8, 6, 2, 1, 3 \}$$

$$\{ 3, 3, 1, 1, 7, 7, 8, 8 \}$$

$$\rightarrow pf[] = \{ 3, 3, 1, 1, 7, 7, 8, 8 \}$$

$$[3, 6] \rightarrow 8 - 1 = 7$$

```

pf[N]
pf[0] = arr[0]
for(i=1 ; i<N ; i++) {
    if( i*102 == 0) {
        pf[i] = pf[i-1] + arr[i]
    }
    else {
        pf[i] = pf[i-1]
    }
}

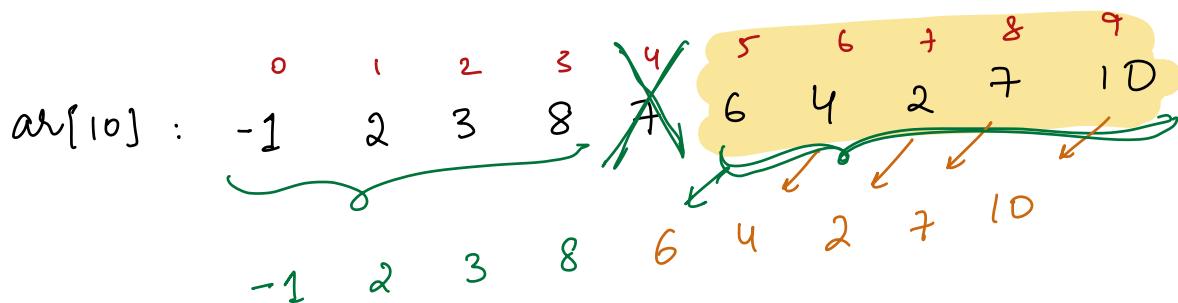
```

pfodd[i] → sum of odd ele from [0 to i]

arr[] → 2 4 3 1 5

pfodd[] → 0 4 4 5 5

// write the code for this



pos 4 by $\rightarrow 6$
pos 4 after $\rightarrow 5$

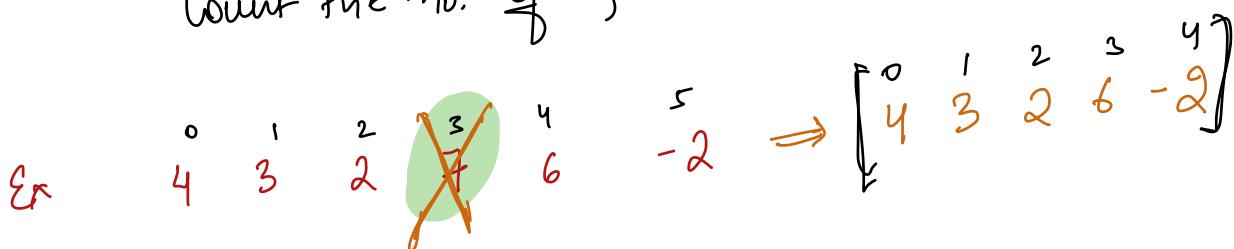
pos of 6 $\rightarrow 5$
pos of 6 later $\rightarrow 4$

Google media.net Special Index.

An index will be called a special index if after deleting that index.

$$\text{Sum of even index value} = \text{Sum of odd index values}$$

Count the no. of special index.



Delete 0th

0	1	2	3	4
3	2	7	6	-2

sumOdd $\Rightarrow 8$
sumEven $\Rightarrow 8$

YES.

Delete 1st

0	1	2	3	4
4	2	7	6	-2

sumOdd $\Rightarrow 8$
sumEven $\Rightarrow 9$

NO.

Delete 2nd index

0	1	2	3	4
4	3	7	6	-2

sumOdd $\Rightarrow 9$
sumEven $\Rightarrow 9$

YES.

arr[14] : 0 1 2 3 4 5 6 7 8 9 10 11 12 13

sum of even index ele
& sum of odd index ele.

check for this

sumEven [0 4]

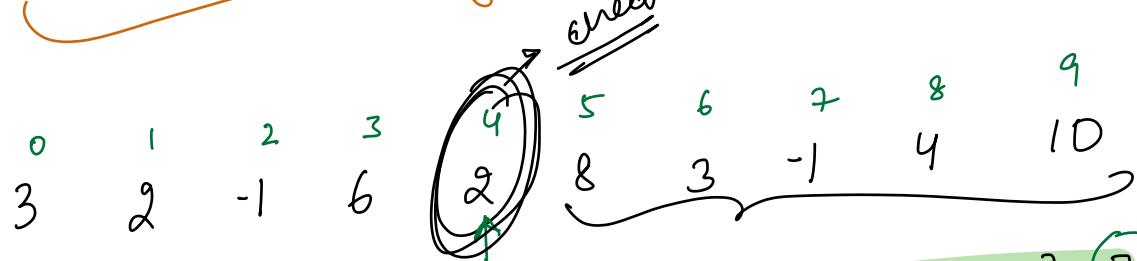
sumOdd [0 4]

sumEven [6 13]

sumOdd [6 13]

$$\text{sumEven} \rightarrow \text{sumEven}[0 \ 4] + \text{sumOdd}[6 \ 13]$$

$$\text{sumOdd} \rightarrow \text{sumOdd}[0 \ 4] + \text{sumEven}[6 \ 13]$$

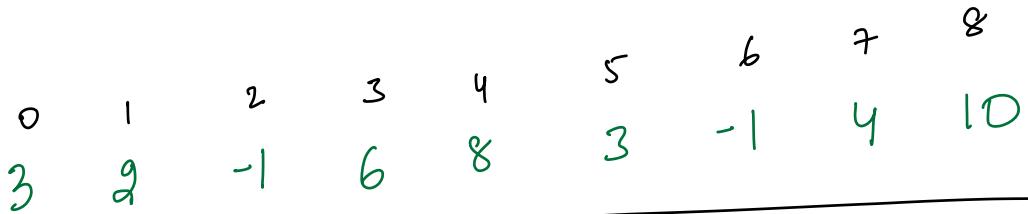


sumEven [0 3] = 2

sumOdd [0 3] = 8

sumEven [5 9] = 7

sumOdd [5 9] = 17



sumEven [0 8] → 19

sumOdd [0 8] → 15

0 1 2 ... i-1 **i** i+1 i+2 ... N-1

sumEven [0 i-1]

sumOdd [0 i-1]

sumEven [i+1 N-1]

sumOdd [i+1 N-1]

sumEven [0 i-1] + sumOdd [i+1 N-1]

=

sumOdd [0 i-1] + sumEven [i+1 N-1]

Then i is your special index.

Pseudocode

① pfEven [] → O(N)
 ② pfOdd [] → O(N)

③ cnt = 0

for(i=0; i < N; i++) {

// check if the index is special or not → O(1)

total_even_sum

= sumEven [0 i-1] + sumOdd [i+1 N-1] → O(1)

total_odd_sum

= sumOdd [0 i-1] + sumEven [i+1 N-1] → O(1)

. if (total_odd_sum == total_even_sum) cnt ++

}
 return cnt

$$TC : O(N) + O(N) + O(N) \\ = O(N)$$

$$SC : O(N) + O(N) = O(N)$$